応用分子動力学計算プログラム

RYUDO

取扱説明書

2003/6/19：初版
2004/5/13：Ver.0.85 対応
2004/5/29：Ver.0.86 対応
2004/6/8：Ver.0.87 並記
2004/6/9：Ver.0.87 専用化
2004/8/23:Ver.0.88 対応
2004/9/8:Ver.0.89 対応
2004/9/21:Ver.1.00 対応
2004/11/22:Ver.1.03 対応
2005/1/20:Ver.1.04 対応
2005/2/7:Ver.1.05 対応
2005/6/4:Ver.1.06 対応
2005/10/19:Ver.1.07 対応
2005/12/24:Ver.1.08 対応
2006/4/5:Ver.1.09 対応
2006/4/8:Ver.1.10 対応
2006/4/19:Ver.1.11 対応
2006/4/19:Ver.1.12 対応
2006/5/27:Ver.1.13 対応
2006/9/14:Ver.1.15 対応
2007/2/10:Ver.1.16 対応
2007/4/16:Ver.1.17 対応
2007/10/24:Ver.1.18 対応
2008/1/15:Ver.1.19 対応
2008/2/5:Ver.1.20 対応
2008/9/14:Ver.1.21 対応
2009/2/13:Ver.1.22 対応
2009/4/15:Ver.1.23 対応
2009/5/20:Ver.1.24 対応
2009/9/18:Ver.1.25 対応
2009/11/27:Ver.1.26 対応
2010/10/19:Ver.1.27 対応

No.20110310

目次

注意：

以下の文中の各種定義ファイルの書き方において、項目間の空白（スペースやタブ）の存在は非常に重要です。項目の間が開いている箇所には、必ず空白を入れるようにしてください。

1. はじめに

近年、計算化学は急速な発展を遂げ、効果的な開発研究には欠かせないものとなりつつあります。特に分子動力学(MD)法は、原子の熱運動や拡散など、時間経過に伴う構造の変化などを観察することができるため、比較的多用される計算手法の一つです。

「RYUDO」は、この MD 計算を有機分子から無機結晶系までの様々な分野に適用し、より有用な知見を得ていくために開発した計算プログラムです。MD 計算の基本的な機能のほかに、剪断場や結晶成長などの様々な応用シミュレーションを行う機能を実装しました。RYUDO の主な特徴は以下の通りです。

- 異なるポテンシャル関数を任意に組み合わせて、複雑な混合系を計算することができます。
- 結合や分子所属の情報を保持し、結合性や分子内にポテンシャルを適用できます。
- 圧力制御は、計算セルサイズのスケーリングで実現しています。ここで、任意の軸だけは変動しないように指定することもできます。
- 温度制御は、各原子の速度スケーリングで行います。また、設定温度一定のほか、定期的に上昇あるいは下降させるシミュレーションを行うことができます。
- 元素と表示方法および原子性質を別々に指定することができます。これにより、同性質の原子に複数の表示方法を与え、サイトなどの識別を容易にしたり、逆に異なる性質の原子に同一の表示方法を与え、性質が微妙に異なり多数の元素を設定している場合でも見やすく表示することができます。
- 任意の元素を常に一定方向に一定速度で移動させることができます。これにより、固体間に挟まれた潤滑有機分子などの剪断場における挙動の観察などが可能になります。
- 任意の元素に常に一定方向で一定の大きさの力を加え続けることができます。これにより、外力による構造の破壊過程などを観察することができます。
- 計算中の系内に、任意の速度をもつ任意の分子原子を、複数個同時に、指定時間間隔で 出現させることができます。これにより、結晶表面の成長過程など計算することができます。
- 系内の任意の領域に移動した原子を自動的に削除させることが出来ます。これにより、表面から真空中に蒸発した原子や、結晶成長過程で跳ね返ってしまった原子を取り除いたり、あるいは原子の 出現機能と組み合わせて分子線照射過程のシミュレーションが可能です。
- 系内に高温領域と低温領域設定して2種類の温度制御を行い、熱エネルギーの伝搬を再現できるようにすることで、熱伝導過程のシミュレーションを実現しています。
- 表面構造の安定維持には、各原子の座標を完全に固定する既存の方法の他、柔らかく初期座標に拘束することで有限の温度を持つことができる固定方法も実現しました。
- 想定した電場方向に基づき、各原子へその電荷に応じた力を加えることで、電場下のシミュレーションを行うことも可能です。

フリーソフトウェアについて

本プログラムは、フリーソフトウェアとして配布しています。どなたも無料でご利用になれますが、著作権（著作権法に定められている著作者の権利）は放棄していませんのでご注意ください。

- 再配布は自由ですが、基本的に入手時と同じ形式で行ってください。実行形式ファイルを付けるなど、構成を変えて配布する場合は、その旨明記して行ってください。
- 著作者の許可無く販売あるいは販売物へ添付することを禁じます。必要な場合はあらかじめご相談ください。
- プログラムの改変は自由ですが、その再配布にあたっては改変点と責任者を明記してください。また、改変物の販売については、上述の通り制限しますのでご注意ください。
- 本プログラムの記述の一部を他のプログラムの開発に利用した場合、その配布や販売に関しては自由です なお、「一部」の範囲については、常識的な判断をお願いします。
- このプログラムを使用した結果については、著作者は一切の責任を負いません。全てご自分の責任においてご利用ください。
- サポートの保証はありません。各種お問い合わせには出来るだけ対応させていただきますが、問題点は基本的にご利用になる方ご自身で解決してくださるようお願いいたします。
- プログラムの改良に関するご提案は歓迎いたします。新機能、新ポテンシャルについては、具体的な実装方法についてもお教え頂けるとありがたいです。

つたないプログラムですが、少しでも諸兄のお役に立てれば幸甚です。

著作者: 三浦　隆治
連絡先: ryujim@d3.dion.ne.jp, miura@aki.che.tohoku.ac.jp

※　東北大学大学院工学研究科応用化学専攻宮本研究室では、本プロクラムをベースにした「NEW-RYUDO」プログラムを開発しており、現在これを代理店を通じて各種サポート・保証付きで販売しております。ご興味のある方はお問い合わせください。

２．コンパイル

RYUDOの配布をソースファイルで受けた場合は、ご使用の前に、ご利用の環境でコンパイル（実行形式ファイルを作成）することが必要です。

・Linuxの場合

RYUDOのソースファイル用にディレクトリを作成し、そこに全てのファイルを展開します。そのディレクトリでmakeコマンドを使ってコンパイルしてください（基本的にコマンドラインで「make」を実行するだけで完了します。オプションなどは不要です）。

コンパイルが終了すると、そのディレクトリに「ryudo」という名称の実行形式ファイルが作成されます。必要に応じて~/bin/ディレクトリにコピーするなどしてお使いください。

なお、開発環境（gccなど）がインストールされていない環境ではコンパイルできませんのでご注意ください。

・その他のUNIXの場合

RYUDOのソースコードはANSI準拠になっていますので、基本的にccなどでもコンパイルできます。ソースファイルを展開したディレクトリのmakefileを環境に合わせて編集してください（例えば、CC=gccをCC=ccに変更し、コンパイルオプション部分（OPT=-c -ansi -Wall）も適宜修正または削除してください）。編集が済みましたら、makeコマンドでコンパイルしてください。

・MS-Windowsの場合

現在、下記の環境において、コンパイルと動作を確認できています。

MS-Windows 2000 Professional ＋ Visual C++ Ver.6.0 Service Pack 6 (RYUDO Ver.1.21)
MS-Windows XP Professional ＋ Visual Studio 2008 Express (RYUDO Ver.1.21)
MS-Windows XP Professional ＋ Windows Service for UNIX 3.5 (RYUDO Ver.1.21)
Windows 7 Professional 64bit + Visual C++ 2010 Express (RYUDO Ver.1.27c)

まずはRYUDOのソースコードを適当なフォルダに展開してください。

Visual C++ Ver.6.0の場合、新規作成・「Win32 Console Application」で空のプロジェクトを作り、「File View」タブでRYUDOのソースコードのうちryudo.c, main.c, load.c, save.c, disp.c, bndd.c, pote.c, defi.h, func.h, valu.hを登録してください。pote.cのみ「設定」→「一般」で、「このファイルをビルドしない」にチェックを入れてください。defi.hの #define MSWINS と #define OLDTIMES をそれぞれ1にしてください(Ver.1.19以降)。後、ビルドして実行形式(*.exe)を得ることが出来ます。

Visual C++ 2010 Expressの場合、新規で空のプロジェクトを作成し、ソリューションエクスプローラーのソースファイルにRYUDOソースコードのryudo.c, main.c, load.c, save.c, disp.c, bndd.cを、ヘッダーファイルにdefi.h, func.h, valu.hを、それぞれ既存の項目として追加してください。pote.cやその他ファイルは登録しないでください。defi.hの #define MSWINS と #define OLDTIMES をそれぞれ1にしてください(Ver.1.19以降)。後、ビルドして実行形式(*.exe)を得ることが出来ます。

使用方法はLinuxとほぼ同様です。上記で得られた実行形式と入力ファイル（def.rd、set.rdなど）を同じフォルダに入れ、実行形式をダブルクリックして起動するか、コマンドプロンプトから当該フォルダに移動して実行してください。なお、ダブルクリックで直接起動した場合にも画面表示を確認できるよう、RYUDO終了時にEnterキー入力待ちになります(Ver.1.19以降)。

・その他の環境の場合

RYUDOは基本的にANSI準拠のC言語で記述されていますので、Cコンパイラさえあればほとんどの環境で

コンパイルできると思います。ただし、現在時刻を取得する関数関係は、機種依存になる可能性がありますので、もしこの部分でエラーが出るようでしたら、適宜削除してください（計算の本質には影響しません）。特に計算終了時間予想のためにマイクロ秒単位で時刻を取得する部分の互換性が低いようです（後述）。
　また、ごく一部の環境で、システムで使用している変数とRYUDOで使用している変数がバッティングすることがあります。通常、システム側はユーザーが使用しないような変数名（例えばアンダーバーで始まる変数名）を用いるので、RYUDOで使われるようなごく普通の変数名（英小文字で始まる数文字程度）ではバッティングしないのですが、古いシステムなどでは希に発生することがあるようです（昔のSolarisで事例があります）。その際は、当該する変数名をRYUDO側で変更する必要があります。

・時間関数の互換性について
　RYUDOでは、計算の進行状況の表示中に、現在の時刻を出力しています。このためにC言語の`time`関数を使用していますが、これは比較的互換性が高いので、ほとんどの環境でそのまま動くと思います。（万一、コンパイル時に問題が生じた場合は、時間表示は計算の本質に関係ないので、ソースコードの当該部分を削除するなどして対応してください。）
　一方、計算終了の予想時刻および計算終了までの残り時間を、より正確に表示させるため、RYUDOのデフォルトで、マイクロ秒単位で時刻を取得できる`gettaimeofday`関数を使用していますが、これに関しては互換性が低いようで、古いLinux(おそらくkernel2.2以前など）やMS-WindowsのVisual C++などではコンパイルできません。この場合は、`defi.h`の `#define OLDTIMES` を1にすることで、従来のtime関数を使った予測計算に切り替えることが出来ます(Ver.1.19以降）（ただし、秒単位までしか取得できないので予測精度は落ちます）。

・非結合性の3体および4体ポテンシャルの計算方法について
　結合していない原子間の3体および4体ポテンシャルは、全組合せの計算のために、通常はそれぞれN^3およびN^4のループ処理が必要になります。これは一般に大変負荷が高く、大規模系では実用的な速度で計算することが困難になります。そこでRYUDOでは毎ステップ、2体ポテンシャルの計算時に近距離の原子間をリストアップ（変数テーブルに記録）しておいて、後、このテーブルを参照して3体および4体の組合せを計算する方法を用いています（デフォルト）。このテーブルを近接原子テーブルと呼び、リストアップする距離範囲および1原子あたりのリスト数を、任意に指定することが出来ます（後述のdef.rdの説明を参照）。
　この方法では、3体、4体の計算もN^1のループ処理になるので、大規模系の高速化には大変有効ですが、高速化のために処理が極端に簡略化されているので、もし上述の「1原子あたりのリスト数」以上の原子が「リストアップする距離範囲」にあった場合はリスト漏れを起こし、正常に計算できません。このため、常に十分なリスト数を指定する必要がありますが、これは少なからずメモリを圧迫します。そこで、通常の方法（テーブルを使わずに、N^3ループおよびN^4ループを使って計算する方法）のソースコードも併記しており、コンパイル時に選択することができます(Ver.1.19以降）。もしこの通常の方法で計算したい場合は、`defi.h`の `#define NNBT34B` を0に変更してからコンパイルしてください。これにより計算速度は著しく低下しますが、「距離範囲」の制限なく全ての組合せを計算することができるようになります。（ただし、Ver.1.19では一部で2体のcutoff距離の影響を受けます。Ver.1.19a以降ではこの影響はありません。）

・付属の支援ツール類について
　RYUDOは本体以外にも、各種支援ツール類がソースコードで付属しています（第10章参照）。これらは「`make`」を実行するだけではコンパイルされません（makeではryudo本体のみが作成されます）。
　これら支援ツールを使用するには、第10章に説明する方法で適宜手動でコンパイルしてください（一部は数学ライブラリが必要になりますので、`gcc`でコンパイルする際は `-lm` （マイナス、エル、エム）オプションが必要になります）。

３．入力ファイルの種類

RYUDO の実行には、カレントディレクトリに以下の入力ファイルが必要です。

・def.rd （計算条件定義ファイル）
計算条件（温度、圧力など）、および原子間ポテンシャルなどを記述するファイルです。
テキスト形式です。サンプルファイルを参考に、テキストエディタなどで作成してください。内容の詳細については後述します。
また、既存の他の計算プログラムの入力ファイルを、添付のツールで変換して作成する方法もあります。その方法についても後述します。

・set.rd （初期構造および初期速度定義ファイル）
計算する系に含まれる全原子の初期座標および初期速度を記述したファイルです。座標は部分座標、速度は[km/s]で記述されています。
テキスト形式ですので、手作業で作成することも出来ますが、通常はモデリングソフトなどで作成した構造データファイルから、各種支援ツール類を通して作成します。その方法については後述します。

このほか、結合性ポテンシャルに必要な結合情報を外部ファイルから読み込みたい場合はそのファイルを、その他特別な計算を行う場合はそれぞれ必要なファイルを、あらかじめカレントディレクトリに用意しておく必要があります。

4. def.rd の詳細

def.rd は、基本的にテキストエディタで作成します。
以下にサンプルファイルを示し、その詳細を後述します。
(なお、これはあくまで解説用ですので、パラメーターの数値等は不正確です。)

```
ryudo-def 1.20    (1)
title    benzen on Fe2O3 surface    (2)
step     100000   0.50[fs]  10       (3)
disp     100      1000      5000     0       yz      si      (4)
files    100      1000      1000     0       0       0       (5)
cutoff   1500[pm]  500[pm]   32      (6)
ewald    6        4.14      0.00027[/pm]     (7)
temp     300.0[K]  1.0[K]  10        10      (8)
press    1.000[GPa]   1.000[GPa]     fix     1       (9)
periodic on       (10)
atom     (11)
         1        1         1        0       0       0       -       (11')
         2        2         2        0       0       0       -
         3        3         3        0       0       0       -
         4        4         4        0       0       0       -
index    (12)
         1        C         100      4       0       (12')
         2        H         50       6       0
         3        O         120      5       0
         4        Fe        70       2       0
nature   (13)
         1 12.01  -0.10  ?  LJ   29687.3[kcalA^12/mol]   132.70[kcalA^6/mol]  (13')
         2  1.01   0.10  ?  LJ      71.4[kcalA^12/mol]    32.87[kcalA^6/mol]
         3 16.00 -2.00  ?  Busing  1.629[A]   0.085[A]
         4 55.85  3.00  ?  Busing  1.207[A]   0.080[A]
pair     table    (14)
         0 0 -    LJ        (14')
         0 0 -    Busing
         1 3 -    Morse     12.00[kcal/mol]   1.980[1/A]   1.780[A]
         2 4 -    Morse     24.00[kcal/mol]   1.980[1/A]   2.130[A]
         1 1 b    Morse     120.0[kcal/mol]   2.000[1/A]   1.340[A]
         1 2 b    Morse     116.0[kcal/mol]   1.770[1/A]   1.080[A]
trio     (15)
         3 4 3 -  SW        (15')
         1 1 1 b Angle     90.00[kcal/mol]  120.00
         2 1 1 b Angle     37.00[kcal/mol]  120.00
quad     (16)
         1 1 1 1 b Torsion  12.000[kcal/mol]   1.00   2.00   180.000   (16')
         2 1 1 1 b Torsion  12.000[kcal/mol]   1.00   2.00   180.000
         2 1 1 2 b Torsion  12.000[kcal/mol]   1.00   2.00   180.000
run      (17)
```

ここで赤括弧()は解説用の注釈番号です。実際のファイルに記述するわけではありません。

各行の概要は下記の通りです(詳細はさらに後ろに記述しています)。

(1)　ファイル識別用。必須です。必ずファイルの先頭に書きます。数字は def.rd 書式バージョン番号。

(2)　計算のタイトル。72文字まで。空白文字が含まれていても可。日本語は非推奨。

(3)　左から、計算ステップ数、タイムステップ(Δt)、運動量補正間隔。
(4)　計算の進行状況を画面に表示する方法の設定。計算結果には直接関係ありません。
左から、1行情報、中間情報、詳細情報、キャラクタグラフィクスの、各表示 step 間隔、
さらに、キャラクタグラフィクスの方向と、表示用単位系(si または mx)の指定です。
(5)　計算結果ファイルの出力 step 間隔の設定。
左から、end.rd、val.rd、pos.rd、vel.rd、for.rd、tmp.rd の出力間隔。
(6)　cutoff 距離と近接原子テーブルの設定。左から、2体ポテンシャル用 cutoff 距離、
3体以上のポテンシャル用 cutoff 距離、および近接原子テーブルの大きさ。
(7)　Ewald 法の設定。通常はこの値のまま使用してください。3次元周期境界条件専用。
(8)　温度制御の設定。左から最終目標温度、昇温速度、制御 step 間隔、平均算出 step 間隔。
(9)　圧力制御の設定。左から、x方向、y方向、z方向、制御 step 間隔。
x、y、またはz方向の圧力を「fix」と書くと、その方向は軸長が固定され、制御されなくなります。
(10)　周期境界条件の設定。on で3次元、off で無し。1、2、3 でそれぞれ1次元、2次元、3次元となります。
(11)　各元素の設定。set.rd 中の元素の種類数だけ必要。
(11')　左から、元素番号、表示方法、性質、局所圧力、(未使用)、準固定、フラグ。
(12)　表示方法の設定。atom 設定で指定した数だけ必要。
(12')　左から、表示方法番号、表示名称、表示半径、表示色、表示形状。視覚化専用です。
(13)　各元素の性質の設定。atom 設定で指定した数だけ必要。
(13')　左から、性質番号、質量、電荷、半径(未使用)、パラメータ種、パラメータ1、2・・・。
注:「パラメータ」は、もし pair 部以降で省略された場合に、デフォルトとして使われる値です。
pair 部以降のポテンシャル指定で値が省略されなかった場合は無意味になります。
(14)　2体ポテンシャルの設定。「table」と書くと、ポテンシャル関数のテーブル化近似を行います。
(14')　左から、元素の組合せ(2項)、フラグ、ポテンシャル名、必要なパラメータ・・・。
元素の組合せは、性質番号で指示します。「0 0」は「全組合せ」の意味になります。
フラグは通常「-」にします。パラメータを省略すると、代わりに性質設定で記述した値が使われます。
(15)　3体ポテンシャルの設定。テーブル化はできません(table を記述しても無効)。
(15')　左から、元素の組合せ(i-j-k)、フラグ、ポテンシャル名、パラメータ1、2・・・。
組合せはjが中心原子。フラグは2体と同じ。パラメーターは省略不可。
(16)　Torsion 型4体ポテンシャルの設定。テーブル化はできません(table を記述しても無効)。
(16')　左から、元素の組合せ(i-j-k-l)、フラグ、ポテンシャル名、パラメータ1、2・・・。
組合せはiとlが端。フラグは2体と同じ。パラメータ省略不可。
(17)　定義終了の意味。RYUDO はここまで読み込むと、計算を開始します。

各行において、各項目の間は、1つ以上の空白(スペースまたはタブ)で区切ってください。桁位置は気にしなくてよいです。1行あたりの最大文字数は 255 です。1行に書くべき内容を、途中で切って2行にしたりしないでください(正常に読み込まれません)。

各行において、先頭に空白を入れては行けません。必ず行頭からキーワードを書いてください。ただし、上記で、番号にダッシュ(')の付いた行は従属行ですので、逆に必ず空白を行頭に入れてください。

各数値についてはカギ括弧 [ ] 内の単位記述にご注意ください。RYUDO はこの部分を読んで単位を自動換算します。正しい単位を記述してください。
2 乗などはべき記号(^)を使い「^2」のように記述します。実数も可(0.5 乗など)。
オングストローム(Å)はアルファベットの A で代用します。

μ（マイクロ）はアルファベットのｕで代用します。
なお、単位記述を省略すると、デフォルトの単位が適用されますが、できるだけ記述した方が確実です。また、一部に単位を使えない（単位が固定されている）パラメーターもあります。単位の詳細については本章末の「参考1」を参照してください。

各行は先頭にコロン「：」を入れることでコメントアウトすることが出来ます。
ちなみに、

temp 行をコメントアウト or 削除すると、エンタルピー一定

press 行をコメントアウト or 削除すると、体積一定

の計算になります。

空白行（何も書かれていない行）を置かないでください。RYUDO が誤作動する可能性があります。見やすさなどのために行を空けたい場合は、必ずコメント行（先頭にコロン「：」を付ける）にしてください。

各行の詳細は下記の通り。

(1) ファイル識別 (ryudo-def)

このファイルが RYUDO 用の定義ファイルであることを示します。def.rd の書式バージョン番号も記述します。def.rd はバージョンによって各項目の意味が変化しますので、必ずこの行を記述し、書式バージョンを明示してください。

なお、ここに記すバージョン番号は、RYUDO 本体のバージョン番号とは異なります。一応、関連性はありますが、必ずしも両者を一致させる必要はありません。ただ、RYUDO のバージョンによっては、古い書式の def.rd を読めないことがありますので、ご注意ください。新規に def.rd を作成したときは、使用する RYUDO のバージョン番号にあわせることをお勧めします。

(2) 計算タイトル (title)

この def.rd で実施する計算のタイトルを記述します。title の後ろに空白を開け、その後ろに72文字以内で自由に記述できます。空白文字を含んでいても構いません。ただし、日本語（2バイト系文字）は正式に対応しておらず、動作不良を起こす可能性がありますので、避けてください。

なお、タイトルが不要の場合は、この行そのものを省略できますが、後の利便性を考えると、あまりおすすめできません。できるだけ何らかのタイトルを付けるようにしてください。

(3) 計算ステップ数など (step)

MD 計算のステップ数、1ステップ当たりの時間間隔、および、運動量補正のステップ間隔を記述します。0ステップは初期状態ですので、最低でも1ステップ以上を指定しないとRYUDO は動きません。1ステップ当たりの時間間隔はデフォルトの単位が[fs]（フェムトセカンド＝10^{-15}秒）です。

・「 1ステップ当たりの時間間隔」は、「タイムステップ」あるいは「積分時間」とも言います。
・運動量補正が不要の場合は、そのステップ間隔（第3項）に0を指定することで無効に出来ます。運動量補正の詳細については、本章末の「参考4」を参照してください。

(4) 画面表示設定 (disp)

計算の進行状況を示す画面表示の設定を記述します。この記述は計算結果に影響ありませんので、適当な値を設定しても問題ありません。ただし、表示が頻繁になりすぎると、実行速度の低下を招きますのでご注意ください。

設定値は、左から順に下記のような内容になります。

- ・1行表示の間隔:温度、圧力、エネルギー、密度、軸調、現在時刻を表示
- ・中間情報の間隔:タイトル、終了ステップ、1ステップの計算にかかった時間、終了予想時刻、および
  各元素(index)ごとのMSDを表示
- ・詳細表示の間隔:終了ステップ、全原子数、内欠陥数、atom数、nature数、index数、
  圧力(x、y、z別)、エネルギー(運動、静電力、短距離力、総計)、
  各元素(index)ごとの原子数、温度、MSDの表示
- ・キャラクタグラフィックスの表示間隔:文字で構造の概要をおおざっぱに視覚化
- ・キャラクタグラフィックスの表示方向:
  xy、xzなど2文字で指定すると、その平面に垂直な向きで表示
  xyzなどと3文字で指定すると、その斜め方向から見る向きで表示
- ・単位系:siと指定すると[pm]や[J]単位、mxと指定すると[A]や[kJ/mol]単位で出力

なお、それぞれの出力間隔に0を指定すると、その表示は行われなくなります。

(5) 計算結果ファイルの出力ステップ間隔 (files)

RYUDOの計算結果を記録する各種ファイルの出力間隔を指定します。間隔の単位はステップです。出力ファイルの詳細については別の章で述べます。

ここに指定する数字は、左から順に、下記のファイルの出力ステップ間隔です。

end.rd	現在の各原子の座標と速度が記録されています。
val.rd	温度、圧力、エネルギー、セルサイズ、MSDなどの履歴です。
pos.rd	各原子の座標の履歴です。
vel.rd	各原子の速度の履歴です。
for.rd	各原子にかかる力の履歴です。
tmp.rd	各原子の運動エネルギー(温度)の履歴です。

end.rd以外の出力ファイルは、ここに指定された間隔で出力が追加されていきます。従って出力回数が多いほど、ファイルサイズが増大します。end.rdは毎回上書きされますので、ファイルサイズは出力回数によらず一定です。

出力間隔に0を指定する、あるいは記述そのものを省略すると、そのファイルは出力されなくなります。ただし、pos.rdやval.rdは後の視覚化と解析に必須となるでしょうし、中断再開のことを考えてend.rdも適当な間隔で出力させておくべきでしょう。そのほかは必要に応じて指定してください。

end.rdとval.rdはテキスト形式ですので、テキストエディタで開いたり、moreコマンドなどで閲覧することができます。そのほかのファイルはバイナリ形式ですので、適宜、処理プログラムが必要です。

なお、val.rdの出力間隔は、計算進行状況記録ファイル「cnd.rd」の経時出力間隔を兼ねています。また、pos.rdの出力間隔と同時に、flg.rdとaxs.rdも出力されます。詳細は出力ファイルを説明する別章を参照してください。

注意:出力間隔を小さくすれば、より詳細な情報が得られますが、そのかわり、ハードディスク容量を消費したり、視覚化解析プログラムで処理に時間がかかったりする恐れがあります。まれに、出力ファイルが大きすぎると、視覚化解析用のプログラムで読み込めないこともあります。ファイルの出力は適宜間引きするようにしてください。目安は一つの計算で100～1000回程度出力されるようにすることです。例えば10,000stepの計算ならば、出力間隔は10～100が適当です。ただし「計算がうまくいかない」「計算開始から短時間で異常終了(温度高温化など)してしまう」などの場合は、各データを毎ステップ出力させて解析することで、原因が分かりやすくなる可能性があります。

(6) cutoff距離の設定 (cutoff)

ここでは、原子間ポテンシャルの cutoff（カットオフ）距離などを設定します。パラメータは左から順に、2体ポテンシャルの cutoff 距離、3体以上のポテンシャル用 cutoff 距離、近接原子テーブルの大きさ、となります。

cutoff 距離とは、原子間相互作用を考慮する距離の上限値です。周期境界条件で計算した場合、原子の組合せは無限遠まで続くので、適当な距離で見切りをつけないと、いつまでたっても計算が終わりません。そこで用いられるのがこの cutoff 距離です。cutoff 距離よりも離れた原子間の相互作用は計算されません。このため、cutoff 距離には、十分に遠方で相互作用がほとんど無視できるような距離を指定する必要があります。もし、あまりにも短い cutoff 距離にしてしまうと、計算精度が著しく落ちるのでご注意ください。一般には 1500[pm]（＝15[Å]）ぐらいが推奨されているようです。なお例外的に、Ewald 法による静電相互作用計算は、カットオフ距離を越えた無限遠近似計算が含まれています。

RYUDO において、実際の2体ポテンシャルの cutoff 距離には、この def.rd の cutoff 行で指定した距離だけでなく、各セルサイズ（a, b, c）の半分の長さも考慮され、この中でもっとも短い値が適用されます。セルサイズが変化する＝圧力制御がある計算の場合は、計算中に自動的に cutoff 距離が変更されます。

「3体以上のポテンシャル用 cutoff 距離」は、非結合性の3体ポテンシャルおよび4体ポテンシャルに適用されます。実際に計算で使用する3体および4体ポテンシャルの効果範囲を考慮し、この cutoff 距離にはそれを十分に超える長さを指定してください（非常に大雑把な例ですと、おおよそ 300～500[pm]ぐらいです）。なお、この cutoff 距離は、直接的には近接原子テーブル（下記）の作成に利用されます。この cutoff 距離より遠い原子はテーブルに記録されません。

> ※近接原子テーブルは非結合性の3体および4体ポテンシャルの計算にのみ用いられます。結合性ポテンシャルについては、結合情報からたぐりますので、近接原子テーブルは使いません。もし、3体以上に非結合性のポテンシャルを使わない場合（たとえば CVFF しか使わない場合）は、近接原子テーブルを作成しない設定にすると、計算時間が短縮されるのでお勧めです。（上記「近接原子テーブルの大きさ」を0にしてください。）

「近接原子テーブルの大きさ」は、テーブル作成時に、1原子あたり周囲の原子をいくつまで記録するかを指定します。デフォルトは 8 ですが、実際には「3体以上のポテンシャル用 cutoff 距離」以内に想定される原子数の倍程度に設定すると良いです（これは熱振動などにより、一時的の想定数以上の原子が cutoff 距離内に入ってくる可能性があるためです。非常に大雑把な例ですと、cutoff 距離が 500[pm]の場合は大きさを 32 以上にすると良いようです）。ただし、あまりテーブルを大きくしすぎると、計算速度の低下とメモリの浪費を招きますので注意が必要です。

RYUDO 実行時、画面にテーブルオーバーフローの警告が表示された場合は、「近接原子テーブルの大きさ」を増やしてください。（この「オーバーフロー」の警告が出たときは、非結合性の3体および4体ポテンシャルが正常に計算できていませんので、ご注意ください。）

> 近接原子テーブルは、ある原子(i)の周囲に存在する他の原子(j)の番号を記録しておくテーブルです。このテーブルは毎ステップ、2体ポテンシャルを計算する際に更新されます。通常、3体や4体の組合せの探索には N^3 や N^4 の計算コストがかかりますが、このテーブル参照形式を用いることで、おおむね N^1 の計算コストで済むようになります（N は原子数）。
>
> 原子 i について、いくつまで近接原子 j を記録するかは、「3体以上のポテンシャル用 cutoff 距離」と、「近接原子テーブルの大きさ」に依存します。この cutoff 距離以下にある原子は全て記録されますが、テーブルの大きさ以上は記憶できず無視されます（オーバーフロー）。高速化のため、テーブルへの記録は、距離の順とは関係無しに行われますので、もしオーバーフローになると、より近い原子が記録されない可能性があります。

(7) Ewald 法の設定 (`ewald`)

静電相互作用（クーロンポテンシャル）を計算するための Ewald 法について設定します。

この行を記述しない（あるいは、コメントアウトする）と、Ewald 法による計算が行われませんので、電荷を持つ原子がある場合は、別途、2体ポテンシャルとして Coulomb ポテンシャルを指定する必要があります（後述）。

また、現在の RYUDO では、Ewald 法は3次元周期境界条件下でしか使用できません。後述する `periodic` の設定で、`off` や `1`、`2` などを指定すると、Ewald 法が使用できませんのでご注意ください。（本来、Ewald 法は1次元や、2次元でも使用できる理論なのですが、開発者の力量不足により、未だ実装できずにいます。もし実装方

法をご存知の方がいらっしゃいましたら、ご教授頂けるとありがたいです。）

この ewald 行で指定する値は、左から順に、逆格子ベクトルの大きさ、α 因子計算用係数1、α 因子計算用係数2、となります。

「逆格子ベクトルの大きさ」は、大きいほど精度が良くなりますが、計算にも時間がかかるようにまります。通常は 6 でも大丈夫ですが、もし精度を良くしたい場合は、11、14、23、29 などを推奨します。30 以上はあまり現実的ではありません。

α 因子は Ewald 法で用いるガウス分布電荷の広がりを規定するもので、計算の収束性に影響します。ここで設定する α 因子計算用係数1･2は、下記の式に基づき α 因子を計算するために用いられます。

$$\alpha \cdot L_{min} = (係数 1) + (係数 2) \cdot L_{min}$$

ここで、L_{min} は、セルサイズ（軸長）a、b、cのうち、最も短いものです。係数1については、採用する逆格子ベクトルの大きさにより推奨値が異なります。（単位は無次元）

逆格子ベクトルの大きさが 6 のとき、係数1＝4.14
逆格子ベクトルの大きさが 11 のとき、係数1＝4.50
逆格子ベクトルの大きさが 14 のとき、係数1＝4.74
逆格子ベクトルの大きさが 23 のとき、係数1＝5.13
逆格子ベクトルの大きさが 29 のとき、係数1＝5.72

係数2については、いずれの場合も 0.027[/A]が推奨値です。

以上から、この行に記述する値としては、下記のものを推奨します。（下に行くほど高精度＆高計算コストです。）（係数2のデフォルトの単位は[/pm]です。）

```
ewald   6    4.14   0.00027
ewald   11   4.50   0.00027
ewald   14   4.74   0.00027
ewald   23   5.13   0.00027
ewald   29   5.72   0.00027
```

以上の説明は、東京工業大学の河村雄行先生の著書「パソコン分子シミュレーション」（海文堂）を参考にしていますので、より詳しくはそちらをご覧ください。

注意：本来、周期境界条件を用いる計算セル内では、電荷の総和は 0 でなければなりません。Ewald 法では特に計算精度に影響しますので、もし総和が 0 でない系を計算しようとすると、RYUDO はエラーを表示して停止します。

注意：結合情報を併用する場合（後述のポテンシャル記述で結合性を使う場合）、Ewald 法による静電相互作用のうち第1項（近距離力）は、結合鎖をたどって4つ以上離れている原子間にのみかかるようになります（すなわち、直接結合している原子間だけでなく、その隣の隣まで＝4体ポテンシャル（Torsion、ねじれ項）の効果が及ぶ範囲までは、直接の静電相互作用がかかりません）。これは CVFF ポテンシャルの定義に基づくものです。もしこれを無視して、結合情報に関係なく全ての原子間で Ewald 法による静電相互作用をかけたい場合は、ewald 行の最終項（第4パラメータ）として、キーワード「bondfr」を記述してください（これを Bond-Free 機能と呼び、Ver.1.23 以降で実装されています）。

(8) 温度制御の設定（ temp ）

この行では温度制御について記述します。RYUDO では、速度スケーリング法による温度制御を採用しています。

この行を記述しない（あるいは、コメントアウトする）と、温度制御を行わなくなりますので、エンタルピー一定の計算になります。

この行で記述する値は、左から順に、最終目標温度、昇温速度、温度制御間隔、平均算出間隔、となります。（温度制御間隔と平均算出間隔は step 数で指定します。）

「最終目標温度」は、文字通り、最終的にこの温度になるよう、系全体の原子速度を調整します。ただし、これは計算が進行していく上での最終的な目標温度であって、計算開始時にはset.rdに記述されている値(通常はset.rd作成時に指定した温度)がとりあえずの目標温度となります。そこから上記「温度制御間隔」ごとに「昇温速度」の分だけ目標温度が変化し、「最終目標温度」に到達するとそれ以上変化しなくなります。「最終目標温度」をset.rdの温度よりも低くし、「昇温速度」に負の値を指定すれば、降温のシミュレーションも可能です。昇温や降温をさせない場合は、set.rdの温度と最終目標温度を一致させてください(「昇温速度」の指定は無効になります。ダミーとして、なにか適当な数値を記述しておいてください)。

「温度制御間隔」は、温度制御処理を実施するstep間隔です。例えば、1を指定すると各stepで温度制御処理が行われ、10を指定すれば10stepおき(10step目、20step目、30step目・・・)に温度制御処理が行われます。通常は10を指定します。間隔を小さくすると温度制御が強くなり、大きくすると弱くなります。

平均算出間隔は、系全体の現在の温度を算出する際に、過去何stepさかのぼって平均を計算するか、を指定します。古典MD計算では、現実に比べて系の原子数がかなり限られているため、系全体の温度は毎stepごとに大きく揺らぎます。このため、現在のstepの瞬間的な温度だけを見ていては、安定した温度制御ができません。そこで、ここに指定する過去step数分の温度を平均し、これを参照して温度制御を行うようにします。推奨値は10(ステップ)ですが、モデルや計算条件によってはもっと長くした方がよいこともあります。

「最終目標温度」の単位はデフォルトがケルビン[K]で、ほかに摂氏[C]も指定できます。昇温速度の単位は[K]固定です(単位を記述しても無効です)。温度制御間隔と平均算出間隔は単位がstepなので単位記述は不要です。

注意:Ver.0.85以前のRYUDOは平均温度の算出方法が違います。このためVer.0.86以降では温度制御間隔の推奨値が10になりました。Ver.0.85以前のdef.rdを流用される方はご注意ください。例えば、現在のバージョンで温度制御間隔を1(=毎step)にすると、過剰制御により温度が不安定になる可能性があります。従来とほぼ同様の温度制御効果を狙うならば、温度制御間隔と平均算出間隔を上記推奨のとおり10にしてください(ただし完全に同等にはなりません)。

(9) 圧力制御の設定 (press)

この行では圧力制御の設定を行います。RYUDOではセルサイズのスケーリングにより圧力を制御しています。

この行を記述しない(あるいは、コメントアウトする)と、圧力制御を行わなくなりますので、体積一定の計算となります。

この行で設定する数値は、左から順に、x方向の目標圧力、y方向の目標圧力、z方向の目標圧力、制御間隔となります。

圧力の単位には[Pa](パスカル)や[atm](気圧)のほか、[bar](バール)や[mHg](メートル水銀)も使えます。デフォルトはパスカル[Pa]です。各方向の目標圧力は別々に設定できます。また、セル長を変動させたくない方向がある場合は、その方向に数値の代わりに「fix」と記述してください。

制御間隔はステップ単位です。通常は1を指定して、毎ステップ制御を行わせます。

(10) 周期境界条件の設定 (periodic)

この行は、周期境界条件の種類を設定します。periodicの後ろに使用したい条件を記述します。

- 数字の1～3で次元を指定します。1次元はx方向のみ、2次元はxy方向の周期境界条件になります。
- onと記述すると、3次元周期境界条件になります。
- offまたは0と記述すると、周期境界条件なしになります。この場合は、真空中にクラスター状態で存在するシミュレーションに相当します。

注意:Ewald法を使用する場合は、3次元周期境界条件を指定してください。もしEwald法を使用していながら、3次元周期境界条件以外を指定すると、RYUDOは停止します。3次元周期境界条件以外で、静電相互作用を

計算したい場合は、pair 部で Coulomb ポテンシャルを使ってください。

(11) 各元素の設定 (atom)

計算モデル中の各元素の基本設定を行います。

この行は、元素の設定の始まりを意味します。パラメーターはありません。具体的な設定は、この行に続く従属行(先頭が空白の行)で行います(下記(12')を参照)。

注意:RYUDO で言う「元素」とは、一般の化学で言う元素とは若干定義が異なり、正確には「粒子の種類」を意味しています。

古典 MD 計算では、例え同じ元素であっても、電荷などが異なる場合は、計算上の都合から、異なる粒子として扱います。例えば、同じ鉄原子であっても、2価と3価では相互作用が変わるため、異なる元素と見なします。

同様に「元素番号」という表記は、一般の周期表にある番号ではなく、計算モデル中に存在する全ての粒子種類の通し番号になります。例えば、酸化鉄モデルなどでは、酸素を1番、鉄を2番と定義したりします。さらに、そこに水分子が存在するならば、水の酸素を3番、水素を4番、などと定義し、「1～4番の、全4種類の元素が存在する」と表現されます。

(11') 元素設定の詳細(従属行)

各元素の設定を記述する行で、atom 行に続けて記述します。各行頭には空白(スペースやタブ)が必要です。また計算に使う元素の数だけ行を記述してください。

ここでは、左から順に、下記の項目を記述します。

元素番号	set.rd に出てくる元素の識別番号です。
表示番号	この元素に適用する表示方法の番号です。
性質番号	この元素に適用する性質(質量や電荷など)の番号です。
圧力番号	この元素が受ける部分圧力(後述)の設定番号です。通常は 0 にします。
(予約)	旧バージョン互換性と将来のための未使用項です。通常は 0 にします。
準固定値	準固定(後述)に用いるエネルギー値です。通常は 0 にします。
フラグ	固定や部分温度などの特殊条件を指定します。通常は「-」にします。

(すなわち、通常は圧力番号以降を「0 0 0 -」と記述してください。)

表示番号は、後述の表示方法の設定で出てくる識別番号を指定します。

性質番号は、後述の元素性質の設定で出てくる識別番号を指定します。

圧力番号は、第7章で説明する特殊計算条件「部分圧力」で使われる設定番号です。部分圧力を使わない場合は「0」と書いておきます。

準固定値は、第7章で説明する特殊計算条件「準固定」で使われるエネルギー値です。準固定を適用しない場合は、1未満の実数値(0 など)を書いておきます。

フラグには下記の意味があります。詳細は後述の特殊条件の計算方法を参照してください。複数指定も可能です(例えば運動量補正無し＋温度制御無しなら「MT」などと記述)。

-	通常原子
X	固定原子(初期座標に固定、原子運動無し＝0K)
M	運動量補正無し(運動量の集計に含まれなくなる)
T	温度制御無し(系全体温度の集計から外れ、速度も制御されない)
H	高温原子(後述の熱伝導計算で使用)
L	低温原子(後述の熱伝導計算で使用)

多くの場合、元素番号＝表示方法番号＝性質番号、として計算できます。煩雑なようですが、この記述方法により、表示は異なるが同じ性質の元素、あるいは、性質は異なるが表示上は一緒、といった特殊な設定が可

能になっています。

これは例えば、同じ性質の粒子でも、存在する場所によって識別したいとき(例えば、基板の酸素と、吸着物の酸素)などに効果的で、性質以下の部分を増やすことなく、表示を切り分けることができます。

例　　atom

```
1 1 1 0 0 0 -
2 2 1 0 0 0 -
3 3 2 0 0 0 -
4 4 3 0 0 0 -
```

この場合、1番の元素と2番の元素は、表示方法は異なりますが、性質は同じ1番となります。これは例えば、MgO と Fe_2O_3 の界面モデルで酸素の性質を一緒にする場合を想定しています。これにより、従来の MD 計算プログラムでは、質量や電荷を4種類、2体ポテンシャルの組合せでは6種類も書かなければならなかったところを、いずれも3種類だけの記述で済むようになります。これはより多くの識別が必要な場合にかなり効果的で、手数とメモリの節約になります。

(12) 表示方法の設定 (index)

各元素に適用する表示方法の設定を行います。

この行は、表示方法の設定の始まりを意味します。パラメーターはありません。具体的な設定は、この行に続く従属行(先頭が空白の行)で行います(下記(12')を参照)。

表示方法は計算結果に影響ありません。元素にどんな名前を付けても結構です。なお、画面表示や val.rd 中の元素毎温度や MSD などは、この表示方法別に集計されますので、これらを細かく分けて解析したい場合は、それぞれに表示方法を設定してください。

(12') 表示方法設定の詳細(従属行)

各表示方法の詳細を記述する行で、index 行に続けて記述します。各行頭には空白(スペースやタブ)が必要です。また使用したい表示方法の数だけ行を記述してください。

ここでは、左から順に、下記の項目を記述します。

表示番号	上述の各元素設定の表示番号に相当
名称	元素の名称、16文字以内、空白不可
半径	視覚化プログラムで使用する表示半径
色番号	視覚化プログラムで使用する色番号
形状	視覚化プログラムで使用する形状番号

半径以降は視覚化プログラム用の記述で、RYUDO の計算自体には影響ありません。なお、視覚化プログラムとしては、RYUGA を想定しています。RYUGA では半径の単位が[pm]に固定されており、単位記述は使えません。

(13) 各元素の性質の設定 (nature)

この行は、表示方法の設定の始まりを意味します。パラメーターはありません。具体的な設定は、この行に続く従属行(先頭が空白の行)で行います(下記(13')を参照)。

(13') 性質設定の詳細(従属行)

各性質の詳細を記述する行で、nature 行に続いて記述します。各行頭には空白(スペースやタブ)が必要です。また使用したい性質の数だけ行を記述してください。

ここでは、左から順に、下記の項目を記述します。

性質番号	上述の各元素設定の性質番号に相当します。

質量　デフォルトは原子単位[au]です。
電荷　単位は[e]で固定です。
半径　省略可能(「?」と記述)。デフォルトの単位は[pm]。

さらにオプションとして、LJ、Busing、および Angell ポテンシャル用のパラメータを書くことができます。半径の記述の後ろに、空白を開けてから、下記のように記述してください。(詳しくはサンプルファイルの記述を参考にしてください。)

・LJ ポテンシャルの場合、LJ と記述した後に、A パラメータ、Bパラメータを書きます。
・Angell ポテンシャルの場合、Angell と記述した後に、n、b、σ、α、E、N パラメータを書きます。
・Busing ポテンシャルの場合、Busing と記述した後に、a, b, c, d パラメーターを書きます。
Busing の c,d パラメータは省略できます(d だけ、もしくは c と d 共に省略できます。
c だけの省略はできません)。

一つの性質に複数のポテンシャルパラメータを記述したい場合は、それぞれキーワードから続けて記述してください(たとえば～ `LJ 1.23 3.45 Busing 1.67 0.08 1.20.3.44` など)。この時、最後のポテンシャル以外ではパラメータを省略できません。パラメータの詳細は別章を参照してください。

半径の記述は、通常は無効ですが、一部のポテンシャルでは平衡核間距離の自動算出に利用されますので、そのようなポテンシャルを使う際には有効です。

注意:電荷と質量以外のパラメータは、pair 部以降のポテンシャル記述のデフォルト値として用いられます。たとえば、もしここで Busing などのパラメータを指定したとしても、pair 部で Busing を記述しない限り、効果がありません(Busing ポテンシャルは掛かりません)。また、電荷についても、Ewald 法を使うか、pair 部で Coulomb ポテンシャルを指定しない限り、効果がありません。

(14) 2体ポテンシャルの設定 (`pair`)

2原子対にかかるポテンシャルを記述します。具体的な設定は、以下に続く従属行で記述します。

この行には、パラメータとして、キーワード「`table`」を指定できます。

キーワード「`table`」を指定すると、2体ポテンシャルのエネルギーおよび力の 関数を数値テーブル化して、計算時間の高速化を図ります。テーブルの精度は 1 pm(=0.01Å)刻みで、その間は適宜補間されます。

テーブル化の範囲は 10 pm(＝0.1Å)から、上述「2体ポテンシャルの cutoff 距離」までです。もし cutoff 距離に大きすぎる値を指定すると、この2体ポテンシャルテーブルも巨大になり、メモリを圧迫しますのでご注意ください。計算中、もし 10[pm]以下の原子対があった場合は、画面に警告を表示すると共に、ポテンシャルが 0 になります。(「2体ポテンシャルの cutoff 距離」以上の場合も、ポテンシャルは 0 になり、相互作用しなくなります。)

このテーブル化は十分に実用的な精度がありますが、もしポテンシャル関数を厳密に計算したい場合は、キーワード「`table`」を削除してください。

なお、テーブル化は2体ポテンシャルだけの機能です。3体および4体ポテンシャルでは使えません。

(14’) 2体ポテンシャル設定の詳細(従属行)

各組合せごとに2体ポテンシャルの詳細を記述する行です。行頭に空白(スペースやタブ)が必要です。また使用したい組合せとポテンシャルの数だけこの従属行を記述してください。

ここでは、左から順に、下記の項目を記述します。

性質番号1　組合せの一端です。
性質番号2　組合せの相手です。
フラグ　ポテンシャルの適用範囲を指定します。通常は「－」(マイナス)。
ポテンシャル名　2体ポテンシャルの種類をキーワード(後述)で指定します。
パラメータ　ポテンシャル関数のパラメータを(複数)記述します。

上述の性質設定(nature 部)で、各性質ごとにパラメータを記述していた場合、ここでのパラメータ記述を省略

できることがあります。詳細は別章のポテンシャルの説明を参照してください。
性質番号は、上述の性質設定(nature 部)の番号に相当します。元素番号ではないことに注意してください。なお、性質番号に0を指定することで、全ての性質を指定したことになります。例えば、

「１０」で、1番と全ての性質の組合せ

「００」で、全ての性質の組合せ

という記述ができます。ただし、これができるのはパラメータを上述の性質設定(nature 部)で記述した場合に限られます(すなわち LJ、Busing、Angell のみ)。通常は必要な全ての組合せを一行ずつ記述してください。
なお、同じ組合せに対し、異なるポテンシャルを重複してかけることができます。たとえば、

```
1 2 - Busing 2.800[A] 0.160[A]
1 2 - Morse 12.0[kcal/mol] 1.980[1/A] 2.880[A]
```

と記述すると、性質1の原子と性質2の原子の間には、Busing と Morse の両方のポテンシャルがかかります。ただし、同じ種類のポテンシャルの場合は、最後に記述したものが有効となります(上書き)。また、上記で Ewald 法を指定した場合、全ての原子間には自動的に静電相互作用が掛かります。
フラグには以下のような意味があります。

-(マイナス)	非結合性(結合無し、または結合鎖で四つ以上離れている原子間)※2
b	結合性(直接結合している原子間)
n	直接結合していない原子間(Ver.1.17a 以降)※2
m	分子内(同一分子内の原子間)※3
o(オー)	分子外(同一分子でない原子間)※3
a	全ての状況(結合情報や分子情報に関係なし)

フラグはどれか一つだけ記述してください。複数指定はできません。
結合性の判断には別途結合情報が、分子内外の判断には別途分子情報が必要です。結合情報と分子情報は set.rd 中に記述します(別章)(結合情報は外部ファイルで指定することも可能です→本章末「参考3」)。
結合情報および分子情報を使わない計算では、デフォルトとして-フラグを記述してください※4。結合情報の無い計算で b フラグを指定するとはエラーになります。分子情報が無い計算で m フラグを指定すると、そのポテンシャルは無効になります。

※1:フラグを数字で複数記述して、適応範囲を詳細に指定することができます。(Ver.1.17a 以降)
0 = 分子外、1 = 分子内で結合鎖四つ以上先、2 = 二～三つ先に結合、3 = 直結(「b」と等価)
複数指定の例: 012 = 直結以外すべて、23 = 直結～3つ先に結合
たとえば、-フラグは 01 と等価、m フラグは 123 と等価になります。

※2:結合情報が無い計算の場合、-フラグや n フラグは、a フラグと同様に全ての原子間に作用します。

※3:RYUDO は基本的に分子間での結合を認めていませんが、結合情報を外部ファイルで指定する場合は分子間の結合を設定できます。この場合、分子内外の判断よりも、結合の判断の方が優先されますのでご注意ください。たとえば、o フラグは結合をたぐって四つ以上離れている原子間にのみ適用されますし、m フラグはそれ以外の全てに適用されます。たとえ分子番号が異なっていても、結合をたぐって三つ以内にある原子間には m フラグのポテンシャルが適用されます。

※4:ここは a フラグでも実質同じことですが、将来の改良時に対応しない可能性があります。この場合、古い def.rd がそのままでは流用できなくなったりしますので、できれば-フラグにしておいてください。

注意:Ver.1.24 以前では、def.rd に b フラグのポテンシャルが一つも無い場合、他のフラグでは結合情報が考慮されません(例えば-フラグや n フラグは全ての原子間にかかり、2 フラグや 3 フラグは無視されます)。Ver.1.25 以降は、結合情報を設定(set.rd 内に記述するか、あるいは bonded 行で外部ファイルを指定)した場合は、必ず結合情報が考慮されます。

その他、キーワードやパラメータなど、2体ポテンシャルの詳細については、別章を参照してください。

(15) 3体ポテンシャルの設定 (trio)

ある原子を中心にした3原子間の角度に依存するポテンシャル、いわゆる3体ポテンシャルを記述します。具

体的な設定は、以下に続く従属行で記述します。

(15') 3体ポテンシャル設定の詳細(従属行)

各組合せごとに3体ポテンシャルの詳細を記述する行です。行頭に空白(スペースやタブ)が必要です。また使用したい組合せとポテンシャルの数だけこの従属行を記述してください。

ここでは、左から順に、下記の項目を記述します。

性質番号1	3原子組合せの端にあたる原子の性質番号です。
性質番号2	3原子組合せの中央にあたる原子の性質番号です。
性質番号3	3原子組合せのもう一端にあたる原子の性質番号です。
フラグ	ポテンシャルの適用範囲を指定します。通常は「-」(マイナス)。
ポテンシャル名	3体ポテンシャルの種類をキーワード(後述)で指定します。
パラメータ	ポテンシャル関数のパラメーターを(複数)記述します。

パラメータはごく一部の例外を除き(後述)、省略することはできません。

性質番号は上述の性質設定の番号に相当します。元素番号ではないことに注意してください。(注:2体ポテンシャルのように、性質番号に全ての意味で0を指定することはできません。)

キーワードやパラメーターなど、3体ポテンシャルの詳細については、別章を参照してください。

フラグは基本的に2体ポテンシャルのものと同様ですが、結合をたぐる判定はありません。

-	非結合性(組み合わせのうち一つでも結合していない場合)
b	結合性(直接結合している原子間)
m	分子内(同一分子内の原子間)
o	分子外(同一分子でない原子間)
a	全ての組合せ(結合情報や分子情報に関係なし)

結合情報を使用しない計算ではデフォルトとして-フラグを記述してください。

bフラグは両端2原子がそれぞれ中心原子に直接結合している3原子にのみ適用されます(どちらか一つでも結合していない場合は、-フラグの適用範囲になります)。

mフラグは3原子が全て同一の分子である場合に適用されます(どれか一つでも違う分子にある場合は、-フラグやoフラグの適用範囲になります)。

※1:フラグを数字で複数記述して、適応範囲を詳細に指定することができます。(Ver.1.17a 以降)

0 = 3原子のうちいずれかが分子外にある場合

1 = 3原子が同一分子内だが、中心原子と他の原子のいずれかが非直結である場合

2 = 中心原子に他の2原子が直結している場合(「b」と等価)

複数指定の例: 01 = 直結以外すべて(ーフラグと等価)、12 = 直結～分子内(mフラグと等価)

注意:Ver.1.24 以前では、def.rd に b フラグ のポテンシャルが一つも無い場合、他のフラグでは結合情報が考慮されません(たとえば-フラグは全ての組み合わせに適用され、2 フラグは無視されます)。Ver.1.25 以降では、結合情報を設定(set.rd 内に記述するか、あるいは bonded 行で外部ファイルを指定)した場合は、必ず結合情報が考慮されます。

なお、一般的に知られている3体ポテンシャルの中には、2体ポテンシャルの項を含むものがあります(SW ポテンシャルなど)。RYUDO では、これらを pair 部と trio 部に分けて記述しなければならないことがあります。詳細は別章のポテンシャルの解説を参照してください。

(16) Torsion 型4体ポテンシャルの設定 (quad)

4つの原子が芋づる状に連続してつながった状態で、前3原子の作る平面と、後ろ3原子の作る平面とのなす角で働くポテンシャルを定義します。いわゆる「ねじれ」を表すポテンシャルで、有機分子で多用されます。なお、この行は記述の始まりを示すのみで、具体的な設定は下記の従属行で記述します。

なお、一般に4体ポテンシャルには Out of Plane(面外角)型もありますが、これについてはここではなく、別途 tetra 部で記述してください(後述)。

(16’) Torsion 型4体ポテンシャルの詳細(従属行)
ポテンシャルの詳細を記述する行です。行頭に空白(スペースやタブ)が必要です。また使用したい組合せとポテンシャルの数だけこの従属行を記述してください。
ここでは、左から順に、下記の項目を記述します。

性質番号1	一方の端にあたる原子の性質番号
性質番号2	性質番号1と3を繋ぐ原子の性質番号
性質番号3	性質番号2と4を繋ぐ原子の性質番号
性質番号4	もう一方の端にあたる原子の性質番号
フラグ	ポテンシャルの適用範囲を指定します。通常は「-」(マイナス)。
ポテンシャル名	Torsion 型ポテンシャルの種類をキーワード(後述)で指定します。
パラメータ	ポテンシャル関数のパラメーターを(複数)記述します。

性質番号は上述の性質設定の番号に相当します。元素番号ではないことに注意してください。(注:2体ポテンシャルのように、性質番号に全ての意味で 0 を指定することはできません。)
フラグは基本的に3体ポテンシャルのものと同様で、「-」「b」「m」「o」「a」※2のいずれかが指定できます。

※1:フラグを数字で複数記述して、適応範囲を詳細に指定することができます。(Ver.1.17a 以降)
0 = 4 原子のうちいずれかが分子外にある場合
1 = 4 原子が同一分子内だが、いずれかが直結していない場合
2 = 4原子が直接連結している(数珠繋ぎになっている)場合
複数指定の例: 01 = 直接連結以外すべて(-フラグと等価)、12 = 直接連結～分子内(m フラグと等価)
注意:Ver.1.24 以前では、def.rd に b フラグ のポテンシャルが一つも無い場合、他のフラグでは結合情報が考慮されません(たとえば-フラグは全ての組み合わせに適用され、2 フラグは無視されます)。Ver.1.25 以降では、結合情報を設定(set.rd 内に記述するか、あるいは bonded 行で外部ファイルを指定)した場合は、必ず結合情報が考慮されます。

ポテンシャルのキーワードやパラメーターなど、Torsion 型4体ポテンシャルの詳細については、別章を参照してください。なお、Torsion 型4体ポテンシャルでは、パラメータの省略はできません。

(17) 計算開始 (run)
この「run」という記述に到達すると、RYUDO はこれまでの記述に基づいて、計算を開始します。
なお、計算が終了すると、この下の行を読み、次の計算を行うことができます。これにより、同じモデルを使って、緩和や、昇温、原子を降らせるなど、複数の計算を連続して実行することができます(後述)。
逆に、RYUDO は計算終了後にこの行以降も読みますので、メモ代わりに変な記述をしたりしないでください。RYUDO の動作が不安定になります。コメントを書きたい場合は、必ず行頭にコロン「:」を入れてください。

参考1:数値単位

各定義に使用する数値の単位は、各数値の直後に[ ]で囲んで自由に指定することができます。
単位としては、以下のようなものが指定できます。
秒[s]、パスカル[Pa]、気圧[atm]、メートル[m]、オングストローム[A]、
ジュール[J]、カロリー[cal]、モル[mol]、ケルビン[K]、摂氏[C]、ミリバール[mbar]、
ステップ[step]、原子単位質量[au]、エレクトロンボルト[eV]、エルグ[erg]、電荷[e]
※オングストロームの記号「Å」は半角英数で入力できないので、代わりにアルファベットの A を使います。
また、桁を表すため、各単位の前に次の文字を入れることができます。
フェムト[f]、ピコ[p]、ナノ[n]、マイクロ[u]、ミリ[m]、センチ[c]、
キロ[k]、メガ[M]、ギガ[G]
後ろにべき記号「^」を付けて、何乗を表すことができます。(乗数は実数や負の値も可)

例　立方メートル = [m^3]

途中に[/]を入れることで、「毎」を指示することもできます。

例　キロカロリー毎モル＝[kcal/mol]

但し、RYUDO の出力については固定されています。たとえば画面表示は上述の disp 行で規定される単位に、ファイル出力はそれぞれ規定の単位（後述）になります。

参考2：Out of Plane 型4体ポテンシャルの記述について

上記サンプルファイル中にはありませんが、RYUDO では Out of Plane 型（面外角型）4体ポテンシャルも使用することができます。ここではその記述方法について説明します。（基本的に上述の Torsion 型4体ポテンシャルと同様です。）

Out of Plane 型4体ポテンシャルとは、3原子からなる平面と、そのうちの1原子（中心原子）と結合する別の原子との結合線がなす角度によって規定されるポテンシャルです。平面に対し、そこからはずれた原子との角度を見ることから、面外角と呼ばれています。（下図参照）

このポテンシャルを記述するには、def.rd 中に tetra 部を作成する必要があります。tetra 部の記述場所としては、quad 部の次を推奨します。

まず、行頭に「tetra」と書き、それ以外には何も記述しない行を作成します（テーブル化はできません。「table」と記述しても無効です）。そしてその下に、空白文字で始まる従属行を記述します。従属行には、左から順に下記の内容を記述します。

性質番号1	中心となる原子の性質番号
性質番号2	性質番号1に結合し、角度を見る原子の性質番号
性質番号3	性質番号1に結合し、平面を構成する原子の性質番号
性質番号4	性質番号1に結合し、平面を構成する原子の性質番号その2
フラグ	ポテンシャルの適用範囲を指定します。通常は「-」。3体と同様。
ポテンシャル名	ポテンシャルの種類をキーワード（後述）で指定します。
パラメータ	ポテンシャル関数のパラメーターを（複数）記述します。

この従属行を必要な組み合わせの数だけ記述してください。また、各項目の注意事項は上述の Torsion 型に準じます。

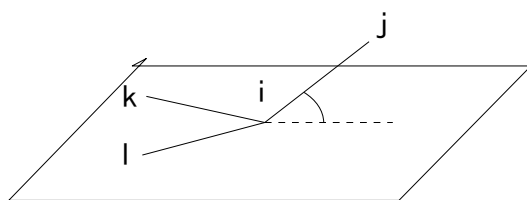



図. Out of Plane の概念図（i,k,l が平面上にある）

参考3: 結合情報について

RYUDO において、結合情報は結合性ポテンシャルを使用する場合に必要となります。結合性ポテンシャルとは、結合していると設定された原子間にのみ働くポテンシャルです（たとえば CVFF の Morse や Angle などで用いられます）。

def.rd において、ポテンシャル記述行のフラグ項※1に「b」を指定することで、結合性のポテンシャルとなります

（逆にフラグ部が「-」の場合は非結合性※2となります）。

※1：pair 部（2体ポテンシャル記述）では各行第3項、trio 部（3体ポテンシャル記述）では第4項、puad 部および tetra 部（4体ポテンシャル記述）では第5項が、フラグ項となります。

※2：非結合性とは、結合情報をたどっても「届かない」、もしくは「4つ以上先にある」を意味します。また、二つ先ないし三つ先の場合は「b」（結合性）にも「-」（非結合性）にも該当しません。これらは CVFF ポテンシャルの仕様に基づいています。

注：RYUDO では、結合情報は単に結合性ポテンシャルを適用するかどうかの判定に用いられます。例えば、同種の組合せ（炭素－炭素など）であっても、「直接結合している」原子間ではフラグに「b」が指定された2体ポテンシャルがかかり、「結合していない（4つ以上先）」原子間ではフラグに「-」が指定された2体ポテンシャルがかかります。たとえば、ある原子間に結合情報が存在したとしても、該当する元素組合せで結合性ポテンシャルの記述が無ければ、相互作用はしません（＝結合したような動きにはなりません）。逆に、結合性ポテンシャル（フラグに「b」）を記述していても、該当原子間に結合情報が無ければ機能しません。

結合情報は通常、set.rd 内の bond 部に記述します。詳細は後述の set.rd の説明をご覧ください。

もし、外部のファイルから結合情報を得たい場合は、def.rd 中に「bonded」行を記述して、そのファイル名を指定してください。

例　　bonded　bondon.dat

なお、bonded は行頭に記述してください（bonded の前に空白を入れないでください）。外部ファイル名は任意です（この例では bondon.dat となっている）。この行の記述位置としては atom 部の前あたりを推奨します。

結合情報ファイルの書式としては、TRIBOSIM プログラムの bondon.dat 形式に準じます。ただし、RYUDO はこの中で、各原子の結合先原子番号のみを読み取ります。その他、結合距離などの情報は無視します。

そのほか、RYUDO 専用の結合情報ファイルとして bnd.rd 形式にも対応します。このファイルの書式は、1原子＝1行として、その原子の番号と結合先原子番号を羅列したものとなります（ちょうど bondon.dat 形式から原子番号情報のみを抜き出したものと同等です）。ただし第1行に「ryudo-bnd 1.00」の様な行が必要です（ryudo-bnd の後の数字は bnd.rd の書式バージョン番号です）。RYUDO はこの「ryudo-bnd」行の存在で bondon.dat 形式と区別しています。

注：set.rd 内の結合情報よりも、bonded 行で指定した外部ファイルの情報が優先されます。

注：外部ファイル（bondon.dat 形式および bnd.rd 形式のファイル中）において指定される原子番号は、原子の通し番号です。原子の通し番号とは、set.rd の part 部に記述された原子の順番のことです（分子番号や分子内番号は関係ありません）。従って例えば 10000 原子ある系では、1 から 10000 までの記述が外部ファイルに必要です。過不足がある場合はエラーで停止します※1。また結合がない原子もかならず1行記述してください（省略できません）。

※1：Ver.1.20i 以前では、set.rd の原子数よりも結合情報が多い場合にのみエラーを表示して停止します。結合情報が足りない場合にはエラーが出ませんが、誤作動します。

注：bonded 行によって結合情報を外部ファイルから取得する方法は暫定機能です。将来の RYUDO では使用できなくなる可能性があります。ご注意ください。

Ewald 法による静電相互作用も、結合情報の影響を受けますので、ご注意ください。

Ewald 法による静電相互作用のうち、第 1 項の近距離力は、結合鎖をたどって4つ以上離れている原子間にのみかかります。すなわち、直接結合している原子間、およびその隣、さらにその隣まで（＝ねじれの4体ポテンシャルの効果が及ぶ範囲まで）は、直接の静電相互作用がかかりません。これは CVFF ポテンシャルの定義に基づくものです。もしこれを無視して、結合情報に関係なく全ての原子間で Ewald 法による静電相互作用をかけたい場合は、ewald 行の最終項（第4パラメータ）として、キーワード「bondfr」を記述してください（Ver.1.23 以降）。

参考4：運動量補正について

現在の計算機の精度は有限（一般的なパソコンで十数桁程度）なので、多原子多ステップの計算では徐々に誤差が蓄積します。これは例えば、真空中に浮かせたクラスターモデルが、何の外力も受けていないのに、徐々

に平行移動するといった形であらわれます。これは現実的ではないですし、拡散係数の算出など後の解析に障害となります。そこでRYUDOでは、系内全原子の重心が計算中に移動しないように、原子の座標を適宜修正する機能を実装しています。このとき原子の速度(厳密には1stepあたりの移動量)も一緒に修正されますので、単純な重心補正とは異なると言った意味で「運動量補正」と言う名称にしています(厳密な定義はわかりません。すみません。各自で調べてください)。

なお、系内で意図的に運動させている原子がある場合、この運動量補正が悪影響を及ぼすことがあります。例えばせん断場のシミュレーションで片側の表面を動かそうとするときに、運動量補正がかかっていると、これを補正しようとするため、動かそうとしている以外の原子が反対側に引きずられてしまうことがあります。また、表面に向かって原子を降下させて結晶成長過程のシミュレーションを行う場合なども、効果原子による重心移動の効果を相殺しようとして表面がせり上がってきてしまうといった現象が発生します。このため、そのような意図的に運動させている原子を運動量補正の監視対象から外したり、あるいは運動量補正そのものを掛けないようにする必要があります。前者の場合、それはdef.rdのatom部において、当該元素のフラグ(第7項)に「M」を指定することで設定可能です。後者の場合、def.rdのstep行において、運動量補正間隔(第3項)を0にすることで設定可能です。特に表面系などで、構造維持のために最下層などが固定されているモデルでは、後者の運動量補正そのものを行わないようにする方法が効果的です。

5. set.rd の詳細

set.rd ファイルは、計算に用いる全原子の初期座標と初期速度を RYUDO に与えることが主な目的です。また、温度制御については、初期の目標温度が、この set.rd に記述されているものになります。

通常、計算モデルには多数の原子が含まれますので、set.rd を手作業で作成するのは困難です。そこで、別途、モデリング用のソフトウェアを用いて、計算モデルの構造ファイルを作成し、全原子の初期座標を決定しておきます。さらに、この構造ファイルをもとに、任意の初期温度にもとづく初期速度を各元素に与えて、set.rd を作成することになります。

各種構造ファイルから set.rd を作成するツールとして、RYUDO には rdset プログラムが付属しています。この rdset プログラムは、構造ファイルの書式として、RYUDO 独自の rst 形式の他、河村先生の MXDORTO などで用いられる xtaldata.dat 形式(ただし空間群は P1 型のみ対応)、Accelrys 社製の Cerius2 で用いられる car 形式および xtl 形式、宮本研究室の COLORS で用いられる xtal.mdy 形式に対応しています。rdset の詳細な使用方法については後述します。

ここで、set.rd のサンプルを用いて、その内容を説明します。
注:このサンプルは、実際に使えるデータではありません。あくまで説明用です。

```
ryudo-set 1.00            (1)
title   SYNTHETIC ZEOLITE        (2)
axis    2.0070e-09  1.9920e-09  1.3420e-09  90.000  90.000  90.000 (3)
cond      300.0     100000    100000    100000    100000       (4)
elem                                     (5)
        1       O       16.00            (5')
        2       Si      28.09
        3       Al      26.98
        4       Na      22.99
bond                                     (6)
        1       2  3                     (6')
        2       1  4
        3       1
        4       2
part                                     (7)
  1 0 0 -  0.3728000  0.0665000  0.7603000  0.1920509  0.0432682 -0.0765181 (7')
  1 0 0 -  0.1272000  0.9335000  0.2603000 -0.0777483  0.1520491  0.1410484
  1 0 0 -  0.6272000  0.5665000  0.2397000 -0.1965809  0.2040482 -0.1833487
  2 0 0 -  0.8728000  0.4335000  0.7397000 -0.0028992 -0.2375698 -0.2019596
  2 0 0 -  0.6272000  0.9335000  0.2397000 -0.2412319  0.0958395 -0.1505995
  2 0 0 -  0.8728000  0.0665000  0.7397000 -0.0479984 -0.0899982 -0.1563883
  3 1 1 -  0.3728000  0.4335000  0.7603000  0.2492714 -0.0195789  0.0475502
  3 1 2 -  0.1272000  0.5665000  0.2603000  0.0348806  0.0959110  0.1968718
  4 1 3 -  0.3095000  0.0621000  0.9239000 -0.0378799 -0.0418711 -0.1733494
  4 1 4 -  0.1905000  0.9379000  0.4239000 -0.0595903 -0.1054811 -0.1130104
end              (8)
```

ここで赤括弧()は解説用の注釈番号です。実際のファイルに記述するわけではありません。

各行の概要は下記の通りです。

(1)　ファイル識別用文。必須です。必ずファイルの先頭に書きます。
(2)　この構造のタイトル。72文字まで。空白文字が含まれていても可。日本語は非推奨。
(3)　セルサイズの記述。軸長(x, y, z)、軸角度(α、β、γ)の順。軸長の単位は[m]。
(4)　温度と圧力の記述。圧力は全圧、x、y、z方向の順。単位は[K]と[Pa]。
(5)　元素の設定。ここから下に従属行を複数記述。
(5')　元素設定の詳細。元素番号、名称、質量[au]の順。
(6)　結合情報の記述。ここから下に従属行を複数記述。不要なら省略可。

(6') 結合設定の詳細。分子内番号、およびその結合先の分子内番号(複数)。
(7) 各原子の記述。ここから下に従属行を原子数分記述。
(7') 元素番号、分子番号、分子内番号、フラグ、初期座標(x,y,z)[-]、初期速度(x,y,z)[km/s]
(8) 終了記号

以下、個別の特記事項です。

・単位について
set.rd では、def.rd のような単位指定([ ]で記述)はできません。各項目の単位は固定されています。
axis 行のセル軸長は[m]、セル角度は[度]です。
cond 行の温度は[K]、圧力は[Pa]です。
elem 部の質量は原子量(au) = [g/mol] です。
各原子の初期座標は、部分座標(軸長で規格化された座標)になっていますので無次元です。
各原子の初期速度は [pm/fs] = [km/s] です。

・セルサイズについて
axis 行の記述で、角度が記述できるようになっていますが、これは将来の拡張用であって、現在の RYUDO では無効です。ただし、将来の互換性を確保するため、必ず「90.00 90.00 90.00」と書いておいてください。

・温度と圧力の記述について
cond 行の温度記述は、計算における初期の目標温度になります。もし def.rd に記述されている温度と異なる場合、set.rd の温度から def.rd の温度に向かって昇温または降温となるシミュレーションを実施します。このときの昇温(降温)速度は、def.rd の temp 行第2項で指定されます。
圧力の記述は、計算条件のデフォルトとなります。def.rd の press 行に値が記述されていれば、そちらが優先されますので、set.rd の記述は無効になります。

・分子番号と結合情報について
結合情報は原子番号(各原子の通し番号)ではなく、各原子に付与された分子内番号で管理されます。このため、同種の分子が複数ある系では、各分子内で共通の分子内番号を使用することで、結合情報を共有することができます。
part 部の各原子の記述において、分子番号とは、その原子が所属する分子の通し番号です。分子に所属しない原子では0にしてください。また、分子内番号とは、その原子が所属する分子内での通し番号です。これは上述のように結合情報に関係します。結合情報を使わない場合は、ここを0にしてください。
なお、分子に所属しない原子(分子番号が0の原子)でも分子内番号を指定(1～)することができます。これを応用して、分子内番号を原子の通し番号とすることができます。この場合、bond 部の結合情報は原子の通し番号で記述できますので、一般の結合情報(bondon.dat など)と同じような設定が可能です。

・フラグについて
各原子の特殊状態を示すのが、part 部の各行第4項のフラグです。
このフラグには以下のような意味があります。

-	通常原子 (「0」と記述しても良い)
d	欠陥原子
a	出現原子

通常原子は、文字通り普通の原子です。

欠陥原子は、計算中は存在しないものとして扱われます。他の原子との相互作用は無効になり、座標が固定されます。これは例えば、構造中から任意の原子を、一時的に取り除いておきたいときなどに利用できます。
出現原子は、後述の原子出現機能を使う場合に必要なフラグです。

・end.rd ファイルについて
現在状態ファイル(end.rd)は、set.rd とほぼ同様の書式になっています。したがって、若干書き換えるだけで、set.rd として流用することができます。(具体的には、先頭行の ryudo-end を ryudo-set に変更し、step 行を削除するだけです。ファイル最後尾の date 行や、各原子記述の末端にある初期座標などは、set.rd として読み込む際に無視されますので、残しておいても大丈夫です。)
なお、end.rd の各原子のフラグは、計算条件(def.rd による)によって上記以外のものが付くことがあります。これをそのまま次の計算の set.rd として用いるには不都合が生じる場合もありますので、その際は適宜フラグを修正してください。(場合によっては付属ツール rdend をご利用ください。)

・構造データ rst 形式について
ryudo 標準の構造ファイルである rst 形式は、この set.rd から cond 行と各原子の初期速度を除いたものになります。すなわち、set.rd は rst 形式の構造ファイルとして用いることができます。end.rd も同様です。

参考1:rdset プログラムの使い方
rdset ［構造ファイル名］［初期温度］［乱数の種］
カレントディレクトリに set.rd が出力されます。すでに set.rd が存在した場合は上書きされますのでご注意ください。
詳細は別章を参照してください。

注意:RYUDO には構造データを構築したり、構造データベースなどの機能はありません。別途論文を参照したり、構造データ作成プログラムを用いるなどして、cartesian 座標系の構造データを作成してください。

注意:原子数上限について(Ver.1.08 以降、Ver.1.19 最新)
結合情報を使う計算を行う場合(def.rd で結合性ポテンシャル(フラグ「b」のポテンシャル)を記述した場合)、デフォルトの RYUDO では取り扱える原子数に上限が発生します。これは2体ポテンシャルの計算において、その2原子が結合で四つ以上離れているかどうかを、フラグテーブルで保存して高速化を図っているためです(RYUDO ではこれを4結合テーブルと呼んでいます)。
フラグテーブルを参照するインデックス変数が unsigned long int 型(32bit)であるため、デフォルトでは 65536 原子で限界になります(途中で N*N の計算があるため)。もし限界を超える原子数の系を計算しようとすると、RYUDO はこれを察知してエラーを出力し、停止します。
フラグテーブルの参照方法を変更することで、メモリ上限いっぱいまで原子数を増やすことが出来ます。これには `defi.h` の `#define FBT_MODE` を 2 に変更して(デフォルトは 1)、RYUDO をコンパイルし直してください。このモードではテーブル数の節約をしませんので、デフォルトより倍のメモリを消費しますのでご注意ください(例えば 65536 原子の場合、約 512MB を消費します)。
テーブルを使わないで計算する方法も選択できます。この場合は `defi.h` の `#define FBT_MODE` を 0 にしてください。メモリは消費しなくなりますが、2原子の組み合わせごとに結合情報をたぐって調べるようになりますので、毎ステップの計算速度は著しく低下します。
(なお Ver.1.07 以前では、上記デフォルトと同様の方法で計算していますが、より単純な(僅かに高速な)テーブル参照方法を採用しているため、およそ4万原子で限界になります。)

６．原子間ポテンシャル

ここではRYUDOで用いることができる原子間ポテンシャルについて説明します。

注意：RYUDOには、パラメータを自動的に決定したり、パラメータデータベースなどの機能はありません。各ポテンシャルのパラメータについては、適用するモデルに合わせて、別途論文を参照したり、他のシミュレーション関連プログラムを用いるなどして決定してください。

６．１．二体ポテンシャル

RYUDOで用いることのできる2体ポテンシャルは下記の通りです。

６．１．１．Lennard-Jonesポテンシャル

基本は下記のエネルギー式になります。（以後、便宜上AB型と呼びます。）

$$E=\frac{A}{r^{12}}-\frac{B}{r^{6}}$$

ここでrは原子間距離です。
このポテンシャルをRYUDOで使用するときは、pair部に'LJ'の行を記述します。

例　　1 1 - LJ 123.23 32.23

ここで、パラメーター部分（'LJ'より右）の第1項がAパラメーター、第2項がBパラメーターです。デフォルトの単位はそれぞれ、[kcalA^{12}/mol] および [kcalA^{6}/mol] です。

nature部で各性質の行末にLJと書いて各性質ごとのパラメータを記述し、pair部ではパラメータ記述を省略して自動的に引用させることができます。nature部に記述するパラメータのデフォルト単位はAが[Jm^{12}]、Bが[Jm^{6}]です。異なる性質の組合せには、AもBも相乗平均が用いられます（下記2式）。

$$A_{ij}=\sqrt{A_i\cdot A_j}$$

$$B_{ij}=\sqrt{B_i\cdot B_j}$$

LJポテンシャルには、εとσを使った次のような式も多用されます。（以後 εσ 型）

$$E=4\varepsilon\left\{\left(\frac{\sigma}{r}\right)^{12}-\left(\frac{\sigma}{r}\right)^{6}\right\}$$

RYUDOではこのタイプのLJを使用することも可能で、その場合、ポテンシャル名に'LJe'を使用します。

例　　1 1 - LJe 0.050e-20 274.0

ここでパラメーター部の第1項が ε、第2項が σ を表し、デフォルトの単位はそれぞれ [J] および [pm] です。

'LJ'と同様に、nature部へ各性質ごとの ε と σ の値を記述して、pair部でのパラメータ記述を省略することができます。nature部に記述するパラメータのデフォルト単位は ε が[J]、σ が[m]です。異なる性質の組合せでは、ε_{ij}に相乗平均、σ_{ij}に相加平均 が用いられます（下記2式）。

$$\varepsilon_{ij}=\sqrt{\varepsilon_i\cdot\varepsilon_j}$$

$$\sigma_{ij} = \frac{\sigma_i + \sigma_j}{2}$$

ただし、次のような Kong の組合せ法則もよく用いられており、RYUDO では'LJk'というポテンシャル名にすることで、この組合せ方法を利用できます。

$$\varepsilon_{ij}\sigma_{ij}^{12} = \frac{\varepsilon_{ii}\sigma_{ii}^{12}}{2^{13}}\{1+(\frac{\varepsilon_{jj}\sigma_{jj}^{12}}{\varepsilon_{ii}\sigma_{ii}^{12}})^{1/13}\}$$

$$\varepsilon_{ij}\sigma_{ij}^{6} = (\varepsilon_{ii}\sigma_{ii}^{6}\varepsilon_{jj}\sigma_{jj}^{6})^{1/2}$$

なお、LJe と LJk の違いは組合せ方法のみですので、nature 部のパラメータを引用しない（pair 部でパラメータ記述を省略しない）場合は、同等のポテンシャルになります。

このほか、LJ ポテンシャルには最安定のエネルギーと距離を直接指定できる下記の形式も存在します（以後 D_0r_0型）。

$$E = D_0\left\{\left(\frac{r_0}{r}\right)^{12} - 2\left(\frac{r_0}{r}\right)^{6}\right\}$$

RYUDO ではこのタイプの LJ を使用することも可能(Ver.1.23 以降)で、ポテンシャル名に'LJ2'を使用します。

例　　1 1 - LJ2 50.00 2.74

ここでパラメータ部分の第1項が D_0、第2項が r_0を表し、デフォルトの単位はそれぞれ [kJ/mol]および [Å] です。

nature 部に性質ごとのパラメータを指定する場合は LJ に続いて D_0と r_0の値を記述します。これらのデフォルトの単位は D_0が[J]、r_0が[m]です。異なる性質の組合わせでは、D_0に相乗平均、r_0に相加平均が用いられます。

注意：パラメータを nature 部で記述し、pair 部で省略する場合は、注意が必要です。

nature 部における LJ パラメータ記述は、AB 型用でも ε σ 型用でも D_0r_0型でも、すべて「LJ」と記述します。これは内部変数が共有されているためです。よって、この段階ではどの LJ 用のパラメータか分からないため、デフォルトの単位は上述のように最も基本的（換算不要）なものになっています。

このことから、一つの def.rd に異なる形式の LJ（たとえば AB 型と ε σ 型）を混在させることは、パラメーター誤用の危険がありますので推奨できません。もちろん十分注意して元素の組み合わせを区別すれば可能ですが、nature 部で AB 型用を指定した性質と ε σ 型用を指定した性質の組み合わせでは pair 部でパラメータを省略できません。

6．1．2．Busing ポテンシャル

あまり聞かれない名前かもしれませんが、これは MXDORTO プログラムなどでイオン結合性モデルの斥力項として使用されてきたポテンシャルで、一般に BMH ポテンシャルなどと呼ばれるものと同等です。（イオン結合性モデルでは、原子間の相互作用はクーロン力が支配的ですが、もしクーロン力だけですと、正電荷と負電荷の原子は無限に近づいてしまうので、これを防ぐために原子核の大きさを表す斥力ポテンシャルが必要になります。MXDORTO プログラムではこの目的のために Busing ポテンシャルを使用しており、RYUDO もその流れを汲んでいます。）

RYUDO の Busing ポテンシャルは下記の式で表されます。

$$E=f_0(b_i+b_j)\exp\frac{a_i+a_j-r}{b_i+b_j}$$

f_0はプログラム内で規定のエネルギー係数で、下記の値を持ちます。単位は[J/m]です。

f_0 = 4.184 / 6.022e23 * 1e3 * 1e10

aパラメータは原子の大きさに関する因子、bパラメーターはポテンシャルカーブの形状に関する因子で、それぞれ長さの次元を持ちます。ここで注意して頂きたいのは、これらのパラメーターは元素ごとに記述することを標準としており、それらを単純に加算することで2元素間の値としていることです（平均ではありません）。よって、Busingポテンシャルでは、nature部へ各元素ごとにパラメーターを記述しておき、pair部ではパラメーターを省略する使い方が標準となります。

nature部への記述では、例えば下記のようにパラメーターを指定します。

例　　　1　16.00　-2.000　?　Busing　1.6[A]　0.080[A]

ここで、「Busing」の後ろの第1項がa、第2項がbパラメーターです。上記の例では単位としてオングストロームを使用していますが、省略した場合のデフォルト単位はいずれも [pm] です。

pair部への記述では、たとえば下記のように記述します。

例　　　0　0　-　Busing

ここでは、元素の組合せごとにパラメータを指定するのではなく、全元素の組合せにnature部の記述に基づいたBusingポテンシャルを定義しています。

なお、もし元素の組み合わせごとに特別にパラメータを指定したい場合は、pair部の記述を以下のようにします。

例　　　1　1　-　Busing　3.344[A]　0.160[A]

ここで、「Busing」以降の第1項がaパラメータ、第2項がbパラメータで、デフォルトの単位はいずれも [pm] です。

注意して頂きたいのは、ここで指定する値は a_i+a_j および b_i+b_j の値だと言うことです。例えば、酸素原子単体のパラメーターが1.6[A]、0.08[A]である場合、酸素－酸素間のパラメーターとしては、3.2[A]、0.16[A]を指定してください。

なお、RYUDOのBusingポテンシャルでは、双極子－双極子相互作用、および、双極子－四重極子相互作用も記述することもできます。その際のエネルギー式は以下のようになります。

$$E=f_0(b_i+b_j)\exp\frac{a_i+a_j-r}{b_i+b_j}-\frac{c_ic_j}{r^6}-\frac{d_id_j}{r^8}$$

ここでパラメーターとしては、双極子－双極子相互作用としてc、双極子－四重極子相互作用としてdが追加されます。また、aおよびbパラメーターと異なり、cおよびdパラメーターは元素の組合せ則として乗算を用いています。

これらcパラメーターおよびdパラメータの記述方法としては、上記nature部またはpair部のa、bに続けて、cおよびdの値を追記するだけでよいです。

たとえばnature部では以下のように記述します。

例　　　1　16.00　-2.000　?　Busing　1.6[A]　0.080[A]　49.09　1.20

ここで、デフォルトの単位はそれぞれ [$kJ^{0.5}pm^3/mol^{0.5}$] および [$kJ^{0.5}pm^4/mol^{0.5}$] です。

また、pair部にパラメータを記述する場合は下記のようになります。

例　　　1　1　-　Busing　3.344[A]　0.160[A]　2498.0　1.82

ここで、デフォルトの単位はそれぞれ [kJ pm^6/mol] および [kJ pm^8/mol] です。

６．１．３. Angell ポテンシャル

BMH ポテンシャルと呼ばれるものと同等です。

基本的に、上述の Busing ポテンシャルと似たような式になっており、パラメーターが2個増えています。

$$E_{ij}=f_0 A_{ij}(b_i+b_j)\exp(\frac{\sigma_i+\sigma_j-r}{\rho_{ij}})-\frac{C_{ij}}{r^6}-\frac{D_{ij}}{r^8}$$

f_0 は Busing と同様に下記の定数になっています。単位は[J/m]です。

f_0 = 4.184 / 6.022e23 * 1e3 * 1e10

A は Pauling 因子と呼ばれるもので、陽イオン間では斥力を大きめに、陰イオン間では小さめにする係数です。b は斥力に、σ は原子の大きさに、ρ はソフトネスに関するパラメーターで、C は双極子－双極子相互作用、D は双極子－四重極子相互作用のパラメーターです。

RYUDO で使用するには、pair 部に Angell キーワードを使って下記のように記述します。

例　　1 1 - Angell　1.0 0.16[A] 3.4[A] 0.16[A] 0.78 0.98

ここで、「Angell」以降のパラメーターは、左から順に、A_{ij}、b_i+b_j、σ_i+σ_j、ρ_{ij}、C_{ij}、D_{ij}、となります。それぞれデフォルトの単位は、[-] [pm] [pm] [pm] [pm^6kJ/mol] [pm^8kJ/mol] です。

Pauling 因子 A は、下記の式で求めることができます。

$$A_{ij}=(1+\frac{z_i}{n_i}+\frac{z_j}{n_j})$$

ここで、z は電荷、n は最外殻電子数を表します。

また、双極子－双極子相互作用 Cij、および、双極子－四重極子相互作用 Dij は、下記の式で求められます。

$$C_{ij}=\frac{3}{2}\alpha_i\alpha_j\frac{E_i E_j}{E_i+E_j}$$

$$D_{ij}=\frac{9}{4e^2}C_{ij}(\frac{\alpha_i E_i}{N_i}+\frac{\alpha_j E_j}{N_j})$$

ここで α は分極率、E は第1イオン化エネルギー、N は全電子数を示します。

RYUDO では、これらを各元素の設定から自動的に算出させることもできます。このためには、 nature 部の各元素毎の性質行の最後に、Angell から始まるパラメーター群を、以下のように記述します。

例　　1 16.00 -2.00 ? Angell 8 0.407 1.420 3.92 1.86 10

ここで Angell 以降のパラメーターは、左から順に、最外殻電子数 n_i、斥力因子 b_i、原子サイズ因子 σ_i、分極率 α_i、第1イオン化エネルギー E_i、全電子数 N_i です。

さらに、pair 部の記述において、Aij、bi+bj、σi+σj、Cij、Dij を省略することで、上述の nature 部のパラメータから自動的に算出されます。（省略するには「?」を記述するか、あるいは行末の項ならば項目そのものを削除します。）

６．１．４. Morse ポテンシャル

RYUDO では、下記の形の Morse ポテンシャルを使用できます。

$$E_{ij}=D_{ij}\{\exp[-2\beta_{ij}(r-r_{ij})]-2\exp[-\beta_{ij}(r-r_{ij})]\}$$

ここで、D_{ij}はエネルギー、β_{ij}はポテンシャルカーブの形状因子、r_{ij}は平衡核間距離です。
Morseポテンシャルには、このほかにも様々な式の形がありますが、その多くは上式に変換することができますので、適宜換算することで他形式のパラメータを流用することができます。

Morseポテンシャルを使用するには、def.rdのpair部にキーワード'Morse'を使って以下のように記述します。
例　　1 1 - Morse 50.0[kcal/mol] 1.980[1/A] 2.74[A]
ここで、パラメーター部分の第1項がエネルギー、第2項がカーブ因子、第3項が平衡核間距離です。デフォルトの単位はそれぞれ [J] [1/pm] [pm] です。
なお、上記パラメータのうち、平衡核間距離については、nature部で半径を指定してある場合に限り、省略する（記述しない）ことができます。元素種類の組み合わせは自動的に算出されます。

Morseポテンシャルは下記の式で表されることもありますが、本質的に上式と同じで、パラメータも一緒になります。

$$E = D_{ij}[1 - e^{-\beta_{ij}(r - r_{ij})}]^2$$

ただし、この式では、エネルギーが平行距離で0，無限遠でD_{ij}となりますので、総エネルギーと圧力制御には違いが出ます。（原子間力はエネルギー変化の微分であるため、上式と全く同一になります。）

6．1．5. 調和振動（絶対結合）ポテンシャル

切れない結合を表すポテンシャルで、エネルギー式は下記ようになります。平衡核間距離よりも近づいても遠ざかってもエネルギーが無限大に発散するため、cutoff距離を超えない限り、原子間には強い相互作用が働き続けます。

$$E_{ij} = \beta_{ij}(r - r_{ij})^2 - D_{ij}$$

ここで、β_{ij}がカーブ因子、r_{ij}が平衡核間距離、D_{ij}が結合エネルギーを示します。UFFなど、論文によっては$\beta_{ij} = 0.5 k_{ij}$、$D_{ij} = 0$としています。
このポテンシャルをRYUDOで用いるには、pair部に下記のように記述します。
例　　1 1 - Harmonic 140.0[kcal/mol] 1.0[kJ/m^2mol] 2.70[A]
ここで、パラメーターは左からD_{ij}、β_{ij}、r_{ij}、となり、それぞれデフォルトの単位は [kJ/mol] [kJ/m^{2}mol] [pm] となります。
なお、上記パラメータのうち、平衡核間距離については、nature部で半径を指定してある場合に限り、省略する（記述しない）ことができます。元素種類の組み合わせは自動的に算出されます。

6．1．6. 長距離振動ポテンシャル

金属結合の相互作用を表すために用いられるポテンシャルです。

$$U = A[\cos(B \cdot r - C) + D] \cdot \exp[-2 \times 10^{10} \cdot (3 \times 10^{-10} - r)] + L \cdot \exp[E(F - r)]$$

ここで、Lは第2項用のエネルギー定数で、6.9478e-21[J]の値を持ちます。また、A～Fがパラメーター、rは対象2原子間の現在距離です。このポテンシャルを使用するには、pair部に以下のように記述します。
例　　1 1 - Vibration 10.0[kcal/mol] 0.78 1.57 1.00 0.08 2.76
ここでパラメーター部は左から順に、A、B、C、D、E、Fパラメーターの値です。デフォルトの単位はJ、m、radに

よるものです。

6．1．7. Gaussポテンシャル

Gauss関数型のポテンシャルです。基本的に単独で用いられることはなく、通常、第1近接距離しか表せないポテンシャルに組み合わせて、ポテンシャルカーブに第2近接以降の小さな谷を作る目的で使用されます。エネルギーは下記の式で表されます。

$$E=-D\exp[-(\frac{r-s}{l})^2\frac{\pi}{K^2}]$$

ここで、Dは結合エネルギー、sは平衡核間距離、lとKはカーブの形状を決める因子になります。

RYUDOで使用するには、pair部に下記のように記述します。

例　1 1 - Gauss 10.0[kcal/mol] 2.80[A] 1.00 0.50[A]

ここで「Gauss」の後ろに記述するパラメーターは、D、s、K、lの順になっており、デフォルトの単位はそれぞれ[J]［pm]［-]［pm]となっています。

なお、同じ元素組合せに複数のGaussを指定しても、後に書いた記述が優先されるため、谷は一つしか作れません。複数の谷で第3近接以降も表現したい場合に対応するため、Gauss2およびGauss3ポテンシャルを用意しました（関数型と使い方はGaussと同等）。これにより、第3および第4近接を表現することができます。なお、今のところ、第5近接以上には対応していませんが、要望があれば作成します。

6．1．8. 数値ポテンシャル

ポテンシャルカーブの形を、複数のエネルギー／距離の点で直接指定する方法です。量子化学計算などで得られた、エネルギー／距離のデータをそのまま投入することができます。点の間は直線補間になりますので、ある程度精密なデータが必要です。また、記述範囲外（最短点より短い側および最長点より長い側）は端のデータがそのまま延長されますので、留意してください。

このポテンシャルを利用するには、エネルギー／距離の一覧を記したファイルが必要です。part部に下記のように記述して、ファイル名を指定します。

例　1 1 - Numeric filename

ここで「filename」の部分がファイル名です。

ファイルの内容としては、1行あたり1点として、左から順に距離[m]とエネルギー[J]を空白を開けて記述してください。また、距離の短い方からの長い方へ、ファイルの先頭から順番に記述してください。

6．1．9. 水分子ポテンシャル

2体力のみで水分子を表現するポテンシャルです。各元素間の組合せ（H-H、H-O、O-O）で、それぞれ異なる式を用います。

H-Hの場合は、下記のエネルギー式を用います。

$$U_{HH}=f_0[\frac{A}{r}+\frac{B}{1+\exp(C\cdot r-D)}-E\exp(F(r-G)^2)]$$

ここで、f_0はプログラム内で規定のエネルギーで、下記の値を持ちます。単位は[J]です。

$f_0 = 4.184 / 6.022e23 * 1e3$
また、このほかのパラメーターは、A=36.1345[A]、B=18[-]、C=40[1/A]、D=82[-]、E=17[-]、F= -7.62177[1/A2]、G=1.45251[A]、となります。
H-O の場合は、下記の式になります。

$$U_{HO}=f_0[\frac{A}{r}+\frac{B}{r^C}-\frac{10}{1+\exp(Dr-E)}-\frac{4}{1+\exp(Fr-G)}]$$

ここで、f_0は同上、A= -72.269[A]、B=6.361637e-92、C=9.19912、D=40[1/A]、E=42[-]、F=5.49305[1/A]、G=12.08471[-]、となります。
O-O の場合は、下記の式になります。

$$U_{OO}=f_0[\frac{A}{r}+\frac{B}{r^C}-\frac{\exp(D(r-E)^2)}{4}-\frac{\exp(F(r-G)^2)}{4}]$$

ここで、f_0は同上、A= 144.538[A]、B=6.862097675e-85、C=8.8591、D= -4.0[1/A2]、E=3.4[A]、F= -1.5[1/A2]、G=4.5[A]、となります。
上述のように、本ポテンシャルは組合せごとに式の形から異なるため、RYUDO では、これら3種類の式を固定で指定する方法をとります。式中のパラメータを任意に変更することはできません。

例　　1 1 - Water 1

ここで、Water 以降のパラメーターとしては、式の種類を指定する数字だけになります。1 のとき H-H、2 のとき H-O、3 のとき O-O を指定したことになります。上述の各パラメータを変更する方法はありません。

6．1．10．Coulomb ポテンシャル

クーロン力、あるいは静電相互作用などと称され、2電荷間の相互作用を表すポテンシャルです。一般によく知られた下記の式で表されます。

$$E_{ij}=\frac{q_i \cdot q_j}{r_{ij}}$$

通常、RYUDO では、静電相互作用は Ewald 法で計算します。これは、静電相互作用が大変長距離に及ぶポテンシャルであるため、他の2体ポテンシャルと同じような cutoff 距離で計算されると、現実とは大きな誤差を生じてしまうからです。（静電気の力が目視できる距離に及ぶことは、下敷きの摩擦実験などで、皆さんよくご存じと思います。）

しかし、現在の RYUDO では、Ewald 法の適用は3次元周期境界条件下に限られています。2次元、1次元、あるいはクラスター状態（周期境界条件無し）のモデルでは、Ewald 法を用いることができません。また、系内の全電荷の総和が0でない系にも適用することができません。（本来、総和が0にならない系はきわめて好ましくないのですが、どうしても計算しなければならない場合もあるかもしれません。）

このような、Ewald 法を適用できない場合に、静電相互作用を計算する方法として、RYUDO では、ここで紹介する Coulomb 型ポテンシャルを採用しています。ただし、Ewald 法に比して、精度が大変低いことをご承知おきください。

この Coulomb 型を用いるには、pair 部に下記のように記述します。

例　　0 0 - Coulomb

nature 部に電荷を記述してある場合は、ここのパラメーター記述を省略できます。（通常、nature 部には電荷も記述しますので、省略して問題ないはずです。）
もし、nature 部に電荷を記述していない場合、あるいは、nature 部の電荷とは無関係に特別な静電相互作用をかけたい場合は、i 原子および j 原子の電荷を並べて記述してください。

例　　　1 2 - Coulomb 1.00 -2.00

ここで各電荷のデフォルトの単位は[e]（電子一個の電荷）です。

このCoulombポテンシャルは、他の2体ポテンシャルと同様のアルゴリズムで計算されるため、2体ポテンシャル用のcutoff距離が適用されます。このため、相互作用の遠くに及ぶ分はcutoff距離でばっさりと切り落とされます。静電相互作用の計算方法としては、かなり思い切った割り切り（近似）が行われていることを念頭に置いてください。

6．2．　三体ポテンシャル

RYUDO で利用できる3体ポテンシャルには以下のようなものがあります。

6．2．1. Unit ポテンシャル

現在角度と目標角度の差分の二乗で規定される、最も単純な3体ポテンシャルの一つです。原子 j を中心として、ベクトル ji、ベクトル jk の角度に依存します。さらに、ij または jk の距離が伸びると効果が弱くなります。このポテンシャルのエネルギーは下記の式で表されます。

$$E=H(\theta-\theta_0)^2\cdot\exp(\frac{r_0-r_{ij}}{s})\cdot\exp(\frac{r_0-r_{jk}}{s})$$

ここで、H はエネルギー、θ_0は平衡角度、r_0は平衡核間距離、s は距離効果因子のパラメーターです。また、θは現在の角度、r_aは現在の ij 間距離、r_bは現在の jk 間距離です。

このポテンシャルを利用するには、trio 部で以下のように記述します。

例　　1 2 1 - Unit 30.0[kcal/mol] 60.0[deg] 1.70[A] 1.0[A]

パラメーター部分の記述は、左から順に、エネルギー、平衡角度、平衡核間距離、距離効果因子で、デフォルトの単位はそれぞれ [kJ/mol] [度] [pm] [pm] です。

6．2．2. SW ポテンシャル

シリコンやゲルマニウムなど、いわゆるダイヤモンド構造を再現するポテンシャルで、2体ポテンシャルの項と3体ポテンシャルの項からなります。詳細は別途文献を参照してください。RYUDO で採用しているエネルギー式はそれぞれ下記の通りです。(U_{ij}が2体、U_{ijk}が3体)

$$U_{ij}=\varepsilon A(\frac{B}{x_{ij}^p}-\frac{1}{x_{ij}^q})\exp(\frac{1}{x_{ij}-a})$$

$$U_{ijk}=\varepsilon\lambda\exp[\frac{\gamma}{x_{ij}-a}+\frac{\gamma}{x_{kj}-a}]\cdot(\cos\theta_{ijk}+\frac{1}{3})^2$$

ここで、x はパラメーター σ で規格化された距離で、$x_{ij}=r_{ij}/\sigma$、$x_{jk}=r_{jk}/\sigma$ となります。また、U_{ij}は、$x_{ij}\geqq a$ のとき 0 に、U_{ijk}は $x_{ij}>a$ または $x_{jk}>a$ のとき 0 となる。

RYUDO でこの SW ポテンシャルを使うには、pair 部および trio 部で、2体及び3体のパラメーターを指定する必要があります。まず pair 部では下記のように指定してください。

例　　1 1 - SW 24.47e-19[J] 0.6022 4 0 1.8 2.0951[A]

ここでパラメーター部は、左から順に、εA[kJ/mol]、B[-]、p[-]、q[-]、a[-]、σ[pm]です。([ ] 内はデフォルトの単位)

また、trio 部では、下記のように記述してください。

例　　1 1 1 - SW 72.9[J] 1.2 1.8 2.0951[A]

ここでパラメーター部は、左から順に、ε λ[kJ/mol]、γ[-]、a[-]、σ[pm]です。([ ] 内はデフォルトの単位)

なお、上記でパラメーターを一切省略すると、デフォルトとしてシリコンのパラメーターが適用されます(上記例の通り)。ただし、この仕様は暫定ですので、将来は無効になる可能性があります。

6．2．3. 水分子ポテンシャル

H_2O 分子構造を再現するための3体ポテンシャルで、東工大の河村先生によって考案されたものです（実際に水分子を計算するには、適切な2体ポテンシャルも必要です。詳細は文献を参照してください）。このポテンシャルのエネルギー式は下記の通りです。

$$U_{ijk}=D\{1-\cos[2(\theta_{ijk}-\theta_0)]\}\sqrt{k_1 k_2}$$

ここで f_k はエネルギー、θ_0 は平衡角度、k_1 および k_2 は ij および ik の距離因子を表します。距離因子は下記の式で表されます。

$$k_i=1/\{\exp[g_r(r_{OH}(i)-r_m)]+1\}$$

このポテンシャルを使用するには、trio 部に下記のように記述します。

例　　1 2 1 - Water 1.1e-11 99.5 7.0[1/A] 1.40[A] 7.0[1/A] 1.40[A]

ここでパラメーター部分は左から順に、エネルギー[kJ/mol]、平衡角度[度]、g_{rij}[1/pm]、r_{mij}[pm]、g_{rjk}[1/pm]、r_{mjk}[pm]、となっています。([ ] 内はデフォルトの単位)

6.2.4. CVFF 型3体ポテンシャル（Angle）

CVFF の角度項に相当するポテンシャルで、ちょうど上述の Unit ポテンシャルから、原子間距離依存の項を取り除いた形になっています。基本的に、結合原子を想定しているため、対象原子間距離が伸びても弱くなりません。非結合性で使用すると、cutoff 距離でいきなり ON/OFF されますのでご注意ください。

このポテンシャルのエネルギー式は下記のようになります。

$$U_{ijk}=H(\theta_{ijk}-\theta_0)^2$$

ここで、H はエネルギー、θ_0 は平衡角度です。

このポテンシャルを使うには、trio 部に下記のように記述します。

例　　1 2 1 - Angle 60.0[kcal/mol] 120

ここでパラメーター部分は左から順に、エネルギー[kJ/mol]、平衡角度[度] となっています。([ ] 内はデフォルトの単位)

６．３．Torsion 型４体ポテンシャル

連続して連なる4つの原子にかかるポテンシャルで、基本的に結合性として用いられ、結合軸のねじれを規定します。具体的には、前3原子からなる平面と、後ろ3原子からなる平面が、なす角 φ を引数とするポテンシャル関数になります。

RYUDO で使用できる Torsion 型（ねじれ型）４体ポテンシャルには下記のものがあります。

６．３．１．CVFF 型 Torsion ポテンシャル

CVFF で用いられている Torsion ポテンシャルで、下記の式で規定されます。

$$E = H[1 + s \cdot \cos(n \cdot \phi - \phi_0)]$$

ここで H はエネルギー、s は倍率、n は 360° 中の繰り返し数、ϕ_0はオフセット角です。

quad 部での記述は下記のようになります。

例　　6 4 1 6 b Torsion 1.423 1.000 3.000 0.000

ポテンシャル名（キーワード）に「Torsion」と指定し、そのパラメータは左から順に、エネルギー[kJ/mol]、倍率[–]、繰り返し数[–]、オフセット角[度]を記述します。

6．4．Out of Plane 型4体ポテンシャル

中心原子と、それに結合する3原子の組み合わせにかかるポテンシャルで、中心原子とそれにつながる2つの原子からなる平面と、残る原子と中心原子との結合線が、なす角 χ によって規定されます。よって面外角ポテンシャルとも呼ばれます。

RYUDO で使用できる Out of Plane 型（面外角型）ポテンシャルには下記のものがあります。

なお、下記の記述中にある元素の組み合わせ i, j, k, l については、先述の通り、i 原子を中心に j, k, l 原子がそれぞれ i 原子に結合しており、結合線 ik および il が作る面に対し結合線 ij のなす角が χ となります。（下図参照）

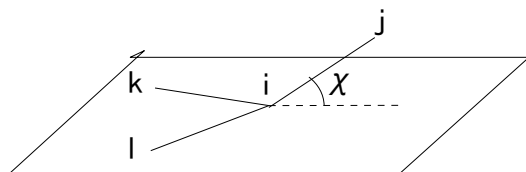



図. Out of Plane の概念図（i,k,l が平面上にある）

6．4．1. 標準型 Out of Plane ポテンシャル

他の MD 計算プログラムで一般的に採用されている角度ポテンシャルです。

$$E=H(1+\cos(n\cdot\chi-\chi_0))$$

ここで H がエネルギー、n が倍率、χ が現在の面外角、χ_0が安定面外角です。H、n、χ_0がユーザー定義のパラメーターです。n は通常 1.0 です。

この関数の tetra 部での記述方法は以下のようになります。

例　　6 4 1 6 b Plane 1.400[kcal/mol] 1.0 0.0

ポテンシャル名（キーワード）に「Plane」と指定し、パラメータはエネルギー（デフォルトの単位は[kJ/mol]）、倍率（単位なし）、角度（デフォルトの単位は「度」）の3つになります。

このポテンシャルは、原子間距離がどれだけ離れても、結合している限り同じ大きさの力がかかります。たとえば結合が伸びるような不安定な計算に適用する場合は注意してください。

6．4．2. CVFF 型 Out of Plane ポテンシャル

CVFF で用いられている Out of Plane ポテンシャルで、下記の式で規定されます。

$$E=H\chi^2$$

ここで H がエネルギーで、χ が面外角です

この関数の tetra 部での記述は以下のようになります。

例　　6 4 1 6 b CVFF 1.400[kcal/mol]

ポテンシャル名（キーワード）に「CVFF」と指定し、パラメータはエネルギー（デフォルトの単位は[kJ/mol]）のみです。

このポテンシャルは、角度 χ が0になる方向に働きます。それ以外の角度は設定できません。また、原子間距離がどれだけ離れても結合している限り同じ大きさの力がかかりますので、結合が伸びるような不安定な計算に適用する場合は注意してください。

注：Ver.1.26 以前のマニュアルでは、標準型の式が誤って E = H (n cos (x − x0))) のように記されていましたが、正しくは E = H cos (n ・ x − x0) でした。Ver.1.27 では式そのものを修正し、上記のようになりました。Ver.1.26 以前とはエネルギーに関する結果が異なってきますのでご注意ください。

６．５．その他（補足事項）

６．５．１．CVFF ポテンシャルについて

CVFF ポテンシャルは、元 Biosim 社（その後の MSI 社、現 Accelrys 社）の分子動力学計算プログラム Discover で使われている有機分子用のポテンシャルです。そのエネルギー式は複数の著名（一部不明？）なポテンシャル関数の和で表されています。

例えば、結合している原子間では、2体項として Morse 型を、3体項として単純な角度依存型、4体項としてねじれ型と面外角型を用いています。（実際にはこの他にも調和振動子型の各種組み合わせがあるようですが、煩雑で効果も薄いようなので RYUDO では実装していません）。

また、分子外、あるいは同一分子内であっても、結合をたぐって4原子以上離れている原子間には、静電相互作用（Coulomb 項）とLJ ポテンシャルが用いられています。

このことから、RYUDO では他のポテンシャルとの記述上の統一を図るため、特に「CVFF ポテンシャル」と独立して実装するのではなく、各項個別に指定する方法を採用しました。

６．５．２．SW ポテンシャルについて

SW ポテンシャルは、そのエネルギー式で既に2体項と3体項に分割されています。そこで RYUDO では、他のポテンシャル記述との統一性を図るため、上述のように、pair 部と trio 部に分けて指定するようにしています。

６．５．３．水分子のポテンシャルについて

水分子は非常に再現が難しい構造の一つで、特に三態を一括して扱えるポテンシャルは、まだ存在しないようです。このため、現在も様々なポテンシャルが考案され、日々研究されているようです。

現在 RYUDO では、水分子を表す方法として、3種類のポテンシャルが選択可能です。

一つは、６．１．９．で示した、2体力だけで水分子の構造を表すポテンシャルです。2体ゆえに計算コストが非常に軽いため、多原子系で効果がありますが、場合によってはうまく水分子の構造を再現できないこともあるようです。（恐らく結合情報なしで計算できるはずですが未確認です）

一つは、６．２．３．で紹介した東工大の河村先生によるポテンシャルです。これは2体項と3体項からなっており、2体項は静電相互作用と Bunsing 型および Morse 型で構成され、3体項は上述の６．２．３．の式で表されます（河村先生の論文を参照してください）。特徴としては結合情報が不要であるため、酸素と水素がばらばらの状態からでもきちんと水分子の構造を再現できます。よって例えば、水分子と−OH 基間で水素原子を交換するような過程も扱えるようです（すみません、未確認です）。

一つは、上述の CVFF によるものです。この場合、非結合性として静電相互作用とLJ 項、結合性として Morse 項と角度項を使用することになります。結合情報が必要になるため、一度決められた水素酸素組は計算中に決して変わりませんが、水分子の構造をかなり良好に再現することができるようです（Discover 中のパラメーターによる）。

なお、いずれのポテンシャルにおいても、水素原子と言う非常に軽い粒子が存在することから、タイムステップ（積分時間）はかなり小さくしなければなりません。CVFF では主に 0.5[fs]、その他の場合は 0.1[fs]などが用いられることが多いようです。

7．特殊計算条件

ここでは、RYUDO で計算できる特殊な条件について、目的別に説明します。

注：下記の各種指定方法で、def.rd 内に新たに single 部の記述を必要とするものがいくつかありますが、これらを組み合わせて用いる場合、single 部は一つにまとめてください。なお、single 部を記述する場所としては、atom 部と index 部の間を推奨します。

7．1．固定、準固定

RYUDO では、元素ごとに原子の固定／非固定（通常）を指定することができます。これは、表面モデルの計算など、構造の一部分を固定したいときに有効です。

原子を固定したいときは、def.rd の atom 部で以下のように記述します。

```
atom
        1     1     1     0     0     0     -
        2     2     2     0     0     0     X
```

この例では、1番の元素は通常（非固定）、2番の元素が固定、となります。

なお、この方法で固定された原子は、初期座標から全く動かなくなります。よって速度は常に0、温度で言うと0[K]相当になります。この場合、固定原子と非固定原子の間に大きな力が生じることがある他、非固定部分の温度が高い場合などは、固定／非固定界面で構造が剥離することがありますので、ご注意ください。

RYUDO では、上述のような 0[K]の固定では問題がある場合に対応するため、原子を初期座標に緩やかに束縛する「準固定」を設定することもできます。この準固定とは、いわば1体ポテンシャルであり、初期座標 r_0と現在座標 r との差を用いて、下記のようなエネルギーで表されます。

$$U = H (r - r_0)^2$$

ここで H はエネルギー係数で[J/m^2]の次元を持ちます。

この準固定を使用したいときは、def.rd の atom 部に下記のように記述します。

```
atom
        1     1     1     0     0     50.0   -
```

ここで、第6項が上記エネルギー係数で、単位は[J/m^2]固定です。この例の場合、1番の元素は、50.0[J/m^2]のエネルギーで初期座標に拘束されます。

この準固定により、有限温度を持ったまま、原子の座標を初期位置に束縛することができますので、高温時の表面モデルなどに効果的です。なお、原子の運動が束縛されることで、温度制御に若干影響が出ますので、ご注意ください。

準固定がかかる下限値は 0.1[J/m^2]です。0.1[J/m^2]以下の値の場合は 0 を指定したと見なし、準固定はかかりません。また、準固定と固定を両方指定した場合は、固定が優先されます。（準固定も慣性運動で計算されているため。）

7．2．熱伝導計算

RYUDO では熱伝導に関する計算が可能です。これは、系内に高温領域と低温領域を設定し、それぞれ を別

個の設定で温度制御する際に、各領域に出入りした運動エネルギーを集計することで、領域間を移動した運動エネルギーを算出するものです。

熱伝導計算を行うには、def.rd の温度制御部分を以下のように記述します。

まず、temp 行について、各項の意味を変えて記述します。

第1項が、低温側の制御目標温度[K]

第2項が、高温側の制御目標温度{K]

第3項は、温度制御間隔[step]　　（通常の温度制御と同じ意味）

第4項は、平均算出間隔[step]　　（通常の温度制御と同じ意味）

さらに、temp 行に続けて次の従属行（行頭に空白が必要）を記述してください。

```
 cond    100    5000
```

ここで、cond の右、第1項は解析用出力ファイル「thc.rd」の出力間隔、第2項は累計開始ステップです。thc.rd の内容については、出力ファイルの解説の章を参照してください。

なお、この記述方法では、高温側および低温側に該当する原子は、atom 部のフラグで指定することになります。例えば下記のように記述すると、

```
atom
        1    1    1    0    0    0    -
        2    2    2    0    0    0    H
        3    3    3    0    0    0    L
```

2番の元素が高温、3番の元素が低温となります。1番の元素は制御無し、すなわち2番の元素から3番の元素へ熱を伝える部分になります。

もし、低温側および高温側原子を、特定の座標範囲で指定したい場合は、上記 cond 行を下記のように記述してください。

```
 cond    100    5000    z   0.1   0.3   0.7   0.9
```

ここで、cond の右、第1項と第2項は上記と同じです。

第3項は熱伝導方向を指定します。x, y, z のいずれかを小文字で記述してください。

第4項以降は、低温側範囲、および、高温側範囲の指定です。部分座標です。

この設定方法を用いたときは、atom 部での H/L の記述は無効になります。

なお、この機能は実際には、指定された範囲内にある原子を、上記「平均算出間隔」で自動的に、低温側または高温側にフラグ設定するものです。よって、平均算出間隔の途中で原子が範囲から出入りしても、平均算出間隔に一致する step までは更新されません（平均算出間隔に一致する step の温度制御後に更新されます）。ただし、平均算出間隔に一致する step であっても、温度制御間隔に一致しない step では更新されません。通常は平均算出間隔を温度制御間隔の整数倍に設定してください。

※temp 行の単位について：プログラム上の問題で、低温側の温度指定（temp 行第1項）にはケルビン[K]のほかに摂氏[C]も使えますが、高温側（temp 行第2項）ではケルビン[K]のみになります（そもそも単位を書いても無視され、強制的にケルビン[K]と判断されます）。

7．3．自動削除（最新 Ver.1.26）

表面から蒸発して真空中に飛び出した原子は、適当な高さに到達したら、系内から取り除きたい場合があります。RYUDO ではこれを「自動削除」機能として実装しています。

この機能を利用するには、def.rd に次のような行を記述します。行頭に空白は不要です。記述場所としては、

atom 部の前を推奨します。

```
autodel    z+    0.8
```

ここで、「autodel」に続くパラメータは、左から順に、方向(x+/x-/y+/y-/z+/z-)と、部分座標です。
指定された部分座標より先に進んだ原子が削除されます。このとき原子の運動方向も考慮されます。すなわち、上述の例の場合、z 軸の＋方向に進む原子で部分座標が 0.8 に到達したものが削除されます。
削除された原子は欠陥状態となり、速度 0 で固定されます。削除以降も end.rd や pos.rd に出力されますが、欠陥状態のため他の原子に影響を及ぼしていません。視覚化の際は欠陥状態を非表示にするとわかりやすいです。
削除対象原子に分子番号(set.rd の各行第2項)がついていた場合(＝分子番号が1以上の場合)、分子単位の削除となり、同じ分子番号をつけた他の原子も同時に削除されます。(Ver.1.26 以降)
削除対象原子に出現フラグ(a、後述)がついていた場合、外されます。よって再出現には使えませんのでご注意ください。(Ver.1.26n 以降)

注:Ver.1.26m 以前では、削除時に出現フラグを削除していなかったため、削除したばかりの原子が出現対象になることがありました。
注:Ver.1.25 以前では、分子の考慮が行われないため、原子単位でしか削除されません。このため、削除範囲に分子がかかった場合、分子の一部原子だけが削除されて計算が不安定になることがあります。また、削除された原子は運動方向に計算セル枠近傍まで移動させられます。分子の場合、削除後に構造が崩れますので、継続計算で削除分子を再利用する場合などはご注意ください。

7. 4. 外力

RYUDO では、指定した元素に属する全原子に、指定した大きさの外力を、計算中継続して掛け続けることができます。
この機能を用いるには、def.rd の single 部に force 行を記述します。下記に例を示します。

```
single
        1        force    1.0e-9[N]     0.0[N]     0.0[N]
```

(ここで single の前に空白を入れてはいけません。一方、force 行は single 部の従属行ですので、行頭に空白(スペースまたはタブ)を入れてください。)
force 行の書式は、左から順に、元素番号(atom 部、すなわち set.rd の記述に準ずる)、「force」＝外力の意、x 方向に掛かる力[N]、y 方向に掛かる力[N]、z 方向に掛かる力[N]、となります。単位を省略した場合、デフォルトでニュートン[N]単位になります。上述の例では、1番の元素に属する原子について、x 方向に $1.0x10^{-9}$[N]の力が常に掛かる様になります。
Ver.1.06 以降ではさらに、計算中にかけ続ける力の大きさを、徐々に大きく、あるいは、徐々に小さくすることができます。この場合、force 行を下記のように記述します。

```
        1   force   1.0e-9[N]   0.0[N]   0.0[N]   1.0e-10[N]   0.0   0.0
```

ここで force 以降は、左から順に、x 方向の初期値、y 方向の初期値、z 方向の初期値、x 方向の増分、y 方向の増分、z 方向の増分、となります。デフォルトの単位はいずれもやはりニュートン[N]となります。増分にはマイナス値も指定できます。この例では、1番の元素に属する原子について、x 方向に、最初(1ステップ目)は $1.0x10^{-9}$[N]の力が、2ステップ目は $1.1x10^{-9}$[N]の力が、以降1ステップごとに徐々に $1.0x10^{-10}$[N]ずつ大きくなる力が、掛かり続けます。

７．５．強制移動

RYUDO では、任意の元素に属する全原子を、周辺の状況にかかわらず、常に一定の速度で強制的に移動させ続けるような計算が可能です。これは例えば剪断場のシミュレーションなどに有効です。

この機能を用いるには、def.rd の single 部に move 行を記述します。下記に例を示します。

```
single
          2    move    -    1.0[pm]    -
```

（ここで single の前に空白は不要ですが、これに続く move 行は single 部の従属行ですので、行頭に空白（スペースまたはタブ）を入れてください。）

move 行の書式は、左から順に、元素番号(atom 部＝set.rd の記述に準ずる)、「move」＝強制移動の意、x 方向の移動速度、y 方向の移動速度、z 方向の移動速度、となります。速度の単位を省略した場合は、デフォルトで pm（＝ピコメーター＝10^{-12}m）単位になります。

また、速度を制御したくない方向（周辺から受ける力に応じて自由に移動させたい方向）については、「－」（マイナス記号）を記述してください（Ver.1.05 以前とは異なりますのでご注意ください）。このとき、[pm]などの単位はつけず、かならずマイナス記号「-」だけを記述してください。

上述の例では、2 番の元素に属する原子について、1ステップごとに、y 方向に 1.0pm ずつ移動します。

上記では速度一定の移動ですが、このほか、加速度（増分）を指定することで、速度（1ステップあたりに移動する量）を計算中に徐々に大きく、あるいは、徐々に小さくすることができます。この場合、move 行は下記のように記述します。

```
          1  move  -   1.0[pm]   -   0   0.1[pm]   0
```

ここで move 以降は、左から順に、x 方向の初期値、y 方向の初期値、z 方向の初期値、x 方向の加速度、y 方向の加速度、z 方向の加速度、となります。デフォルトの単位はいずれもやはり pm となります。もちろん加速度にはマイナス値も指定できます。この例では、1番の元素に属する原子について、x 方向に、最初（1ステップ目）は 1.0pm の速度で移動し、2ステップ目は 1.1pm の速度で、以降1ステップごとに徐々に 0.1pm ずつ速くなっていきます。上限はありませんので、計算が続く限り速度は変化し続けます。

加速に上限を設定したい場合は、加速度につづけて上限値を x、y、z 方向別に記述してください。

```
          1  move  -   1.0[pm]   -   0   0.1[pm]   0   0   5.0[pm]   0
```

この場合、move 以降は左から順に、x 方向の初期値、y 方向の初期値、z 方向の初期値、x 方向の加速度、y 方向の加速度、z 方向の加速度、 x 方向の上限値、y 方向の上限値、z 方向の上限値、となります。上限値に付いてもデフォルトの単位は pm となります。この例では、1step あたり 1.0pm の速度で開始し、以降、毎 step 0.1pm ずつ加速し、5.0pm の速度に到達したところで上限となり、そのまま計算終了まで同じ速度を維持します。

加速度、および上限値に付いて、制御しない方向（初期値に「-」（マイナス記号）のみを記述した方向）については、何を指定しても影響ありませんが、念のため 0 を記述しておいてください。

なお、強制移動が指定された方向については、対象原子が周辺より受ける力の影響は無視されます。これにより、周辺状況がどのような場合でも、必ず指定された速度で移動することになりますが、他原子の異常接近による計算の非常停止（温度が上がりすぎるなど）が発生しやすくなりますので、ご注意ください。

注：Ver.1.10 以前では、強制移動する方向の速度を vel.rd や tvl.rd などで見ると、設定した値と異なることがあります。これは move で補正される前の速度（各原子が受けた力から計算された本来の速度）をそのまま出力しているためです（実際にはこの速度ではなく、ちゃんと move で設定した速度で動きます）。解析の幅を広げるための仕様でしたが、混乱を生じる可能性もあることから、Ver.1.11 以降では move で設定した値がそのまま出るように変更しました。

7．6．部分圧力

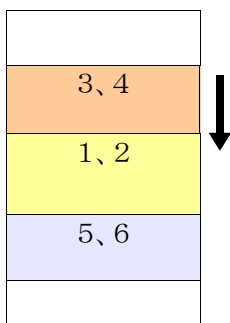
RYUDO では系内の元素ごと(複数まとめても可)に、任意の方向の圧力をかけることができます。これにより、例えば2枚の鉄表面に挟まれた潤滑油分子群を、任意の圧力下で剪断場にかけるようなシミュレーションが可能になります(鉄表面の一方または両方にこの機能を適用します)。

なお、通常の外力のように原子ごとに力を加えると、接触先の形状によって加圧原子団が変形することになりますが、剪断場などのシミュレーションにおいては、このような変形は不都合になる場合があります。そこで、指定された原子団(元素)については、一体で移動させるようにすることで、変形を防いでいます。

この機能を利用するには、atom 部の記述と、single 部の記述の両方が必要です。

まず、atom 部では、部分圧力をかける元素に、第4項で設定番号を記述します。

```
atom
        1       1       1       0       0       0       -
        2       2       2       0       0       0       -
        3       3       3       1       0       0       -
        4       4       4       1       0       0       -
        5       5       5       0       0       0       X
        6       6       6       0       0       0       X
```

この例では、炭素(1番の元素)と水素(2番の元素)からなる分子を、酸化鉄表面(3～6番の元素)で挟んだ状態を想定しています。仮に、3番と5番を酸素、4番と6番を鉄とし、3，4番に部分圧力をかけ、5，6番を固定するものとします。

ここで注目して頂きたいのは、3番と4番の元素の第4項です。通常ここは0ですが、部分圧力をかける場合は、1以上の任意の整数を指定します。3番の元素と4番の元素は同じ圧力を受けて一緒に動かなければなりませんので、同じ番号(ここでは1)を指定します。

次に、single 部の記述を説明します。以下に例を示します。

```
single
        1       press   0.0[GPa]    0.0[GPa]    -1.0[GPa]
```

ここで、従属行の第1項が設定番号(上述の atom 部の第4項とリンクする)、第2項「press」が部分圧力の意、第3項が x 方向の圧力、第4項が y 方向の圧力、第5項が z 方向の圧力です。この例では、z軸方向下向きに1.0[GPa]の圧力が掛かることになります。

ちなみに、RYUDO 内の具体的な処理としては以下のような作業を行っています。

1)圧力方向に垂直なセルの断面積を算出し、これと設定された圧力を乗算して、目的元素(加圧される元素)に属する原子全体に掛ける力を求めます。

2) 目的元素に属する原子が、その他の原子から受ける力を総和して、目的元素全体で受ける力を算出します。
3) 目的元素の全原子数と質量を掛けて、一体で動く全質量を求めます。
4) 目的元素が加圧で受ける力と、他の原子から受ける力を加算し、これと上述の全質量を用いて、目的元素全体の加速度を算出します。
5) 目的元素全体の加速度と、前ステップの速度から、現在の速度を算出します。
6) 目的元素全体の速度を、所属する全原子にコピーします。
7) 後の処理は他の原子と同じです。現在の速度から次の座標が決定されます。

7.7. 原子出現

RYUDO には、set.rd で欠陥（上述）に指定された原子を、任意の間隔で通常原子に戻す機能があります。これにより、あらかじめ set.rd で欠陥として用意していた原子を、計算途中で出現させることができます。これは例えば、結晶成長過程など表面上に原子を次々と落とす計算に利用することができます。

この機能を利用するには、set.rd に出現フラグつきの欠陥原子を準備しておく必要があります。

例
```
1  0  0  -   0.5000  0.5000  0.5000   0.1234  0.2345  0.3456
1  0  0  ad  0.5000  0.5000  0.8000   0.0000  0.0000 -1.0000
```
ここで上の行が普通の原子、下の行が出現フラグ(a)つきの欠陥(d)原子です。
なお、ここで記述する初期速度（右端3項）は、原子出現時に用いられますので、意味のある数値を設定してください。他の原子と同様、左から順に x、y、z 方向で、単位は[km/s]です。

次に、def.rd に single 部を記述して、下記のように指定する必要があります。

例
```
single
         1        appear  1000     500      2
```
ここで、第1項が元素番号、第2項「appear」が原子出現設定の意、第3項が出現開始ステップ、第4項が出現処理間隔、第5項が一回の出現処理で現れる原子数、となります。
この例では、第1元素の出現フラグつき欠陥原子が、計算開始から 1000 ステップ目で2個だけ欠陥フラグをとかれて出現し、さらに 500 ステップおきに2個ずつ出現していきます。出現フラグつき欠陥原子が尽きるまで、500 ステップごとに出現処理が繰り返されます。

また、水分子など、複数の元素からなる分子を出現させたい場合は、下記のように設定します。

set.rd　（3 が H、4 が O）
```
3 0 0 ad 0.000 0.000 0.900 0.000 0.000 -1.000
3 0 0 ad 0.100 0.000 0.900 0.000 0.000 -1.000
4 0 0 ad 0.050 0.000 0.850 0.000 0.000 -1.000
```
（注：速度を一緒にしないと、出現した瞬間に分解する可能性があります。）

def.rd　（3 が H、4 が O）
```
single
         3        appear  1000     500      2
         4        appear  1000     500      1
```
（出現させる組成に基づいて一度に出現する原子数を決めてください。）

分子線照射過程など、表面到達まで初期速度を正確に維持したい場合、温度制御をはずす必要があります。また、大量の原子を出現させて表面上に降下させると、表面構造が運動量補正機能によって原子出現場所方向に移動してしまうことがありますので、場合によっては運動量補正を外すことも必要になります。
これらの指定は、atom 部の第7項で元素ごとに指定可能ですが、最初から系内に存在する原子も影響を受けてしまうため、場合によっては不都合を生じます。そこで、出現させた原子だけを温度制御や運動量補正の対象外にすることができるようにしました。
この機能を利用するには、当該元素の appear 行第6項に、外したい制御・補正の種類を記号で記述してください。「T」ならば温度制御の対象外に、「M」を記述すれば運動量補正の対象外に、「MT」で両方が対象外になります。また、出現後しばらくしたら（たとえば表面に到着して安定したら）制御をかけられるようにするため、対象外にしている期間を第7項に step 数で記述できるようにしました（この期間終了後は atom 部の記述に従うことになります）。

例
```
single
          5       appear  1000   500    2      M
          6       appear  1000   500    2      MT     10000
```
この例では、第5元素は出現した原子について計算終了まで運動量補正が対象外になります。第6元素は出現時に温度制御と運動量補正が対象外になり、出現から 10000step 経過すると atom 部第7項の指定になります。

注意：Ver.1.26m 以前の RYUDO では、上述の自動削除と併用して、出現原子が削除された場合、出現フラグが外れていませんでしたので、次の出現時にこの削除された原子が対象となることがありました。しかしこれは削除時の速度が維持されていますので、すぐに削除されてしまい、結果として正しい出現になりませんでした。
注意：Ver.1.15 以前の RYUDO では、出現させる原子と同元素の原子は、atom 部の記述にかかわらず、すべて自動的に温度制御と運動量補正の対象外となります。出現原子と同じ元素の原子が表面構造などにすでに多数存在する場合、これらの温度制御と運動量補正も対象外になってしまうので注意が必要です。基本的に、出現させる元素と、系内にすでに存在する元素は、別にしてください。
注意：Ver.1.16 以前の RYUDO では、出現用として set.rd に用意する原子は、ただ欠陥フラグ(d)をつけるだけで OK でした。このため、以前の RYUDO で使用していた set.rd を使うと、Ver.1.16a 以降の RYUDO では原子が出現しないことがあります。この場合、あらかじめ目的の原子に出現フラグ(a)をつけてから計算してください。

手作業で set.rd を修正するのはなかなか煩雑ですので、暫定的に「自動で出現フラグをつける」機能を実装しました。これには def.rd のできるだけ後の方で、以下のような行を記述してください。

例 `option appear2`

※option を行頭にしてください（option の前に空白を入れないでください）。
※appear の後ろに出現させたい元素の番号を指定してください（appear と元素番号の間には空白を入れないでください）。上記の例では、元素2の欠陥原子に出現フラグがつくことになります。他の元素の欠陥原子には出現フラグはつきません。
※ここで指定した元素については、set.rd で欠陥(d)となっている原子全てに出現(a)のフラグが追加されます（例えば end.rd において出現前の原子フラグは ad になります）。
※複数の元素を指定したい時は、複数の appear 項を空白を開けて列挙してください。

例 `option appear3 appear4`

※この機能はおまけですので、後のバージョンでは削除される可能性があります。

7．8．垂直ポテンシャル

RYUDO では、仮想的な平面上に垂直に接近する原子(または分子、クラスター)に掛かるポテンシャルを、数値的に指定できる機能があります。これは例えば、数万原子を超えるような大規模クラスターの表面拡散シミュレーションなどに利用できます。

この機能を利用するには、ポテンシャルの数値テーブルを別途ファイルで用意してから、def.rd の single 部に下記のように記述します。

例
```
single
        5      verti   z+      0.2     vpot.dat
```

ここで、第1項が元素番号、第2項「verti」が垂直ポテンシャルの意、第3項が仮想表面の向いている方向(x+、y-など)、第4項が表面があると仮定する座標(部分座標)、第5項がポテンシャルファイル名です。

ポテンシャルファイルの書式としては、上述の Numeric ポテンシャルと同等で、各行に距離[m]とエネルギー[J]を空白を開けて並べて記述します。また表面に近い方から順に遠い方まで記述します。各行間は適宜補間され、近接方向および遠距離方向は適宜外挿されます。

7. 9. 水平ポテンシャル(最新 Ver.1.18)

担体上での拡散など、表面モデルを使うシミュレーションは、表面を構成する原子が多くなるため、MD 法でもかなり時間の掛かる計算となります。そこで RYUDO では、拡散する原子(または分子、クラスター)が表面から受けるエネルギーを、2次元の数値マップで保持することで、表面を構成する原子を省いた拡散シミュレーションを可能にしました。

この機能を用いるには、ポテンシャルの2次元数値テーブルを別途ファイルで用意してから、def.rd の single 部に下記のように記述します。

例
```
single
        5      horiz   z       0.2     hpot.dat
```

ここで、第1項が元素番号、第2項「horiz」が水平ポテンシャルの意、第3項が仮想表面の向いている方向(x、y など)、第4項が表面があると仮定する座標(部分座標)、第5項がポテンシャルファイル名です。

ポテンシャルファイルの書式としては以下のようになります。

- 行頭に「axis」と書き、このファイルで記述する水平面範囲(横と縦)を m 単位で記述します。
- 行頭に「separate」と書き、水平面範囲を横と縦にそれぞれ何点に分けるかを記述します。
- 行頭に空白を書き、各点におけるエネルギーを記述します。左から、横方向の座標番号、縦方向の座標番号、エネルギー[J]です。座標番号は0以上の整数です(例:4点分割ならば0～3)。
- 最後は「end」で終了します。

```
axis      475.33e-12  824.00e-12
separate     4   8
   0   0   -3.9531284599999998e-19
   1   0   -4.6458793400000003e-19
   2   0   -3.9762739000000000e-19
   3   0   -3.8056311400000000e-19

         ...

   1   7   -3.7651346299999999e-19
   2   7   -3.7541889800000002e-19
   3   7   -4.2841242500000001e-19
end
```

ここで「縦方向」と「横方向」の意味については下記のとおりとなります。

- 水平面が x 軸方向に向く場合、「横方向」とは y 方向、「縦方向」とは z 方向になります。
- 水平面が y 軸方向に向く場合、「横方向」とは z 方向、「縦方向」とは x 方向になります。

● 水平面がz軸方向に向く場合、「横方向」とはx方向、「縦方向」とはy方向になります。

実際の計算セルサイズが、上記「axis」で記述した範囲よりも大きい場合、このポテンシャルマップが周期的に繰り返されて埋め合わされます。逆に計算セルサイズがこれより小さい場合、はみ出した範囲は無視されます。

注意：Ver.1.17以前のRYUDOでは暫定仕様のため、水平面の方向としてはzのみでした（single部のhoriz行第3項にxやyを指定しても無効です）。

7．10．組合せ機能

上述の特殊機能を組み合わせることで、様々な応用計算が可能です。

・分子線照射

上述の、原子出現と自動削除を組み合わせることで、例えば、表面モデルに原子あるいは分子を空中から衝突させて、表面原子を熱的に励起するようなシミュレーションが可能です。衝突後の原子ははねかえり、適当な座標に到達すると削除されます。

・加圧剪断場

トライボロジー分野に置いて、加圧された剪断場における潤滑油分子の挙動解析は常に重要な課題です。RYUDOでは、上述の強制移動と部分圧力を組み合わせることで、一定圧力を加えた状態で、一定速度の剪断場を再現することが可能です。

7．11．テキスト出力機能

これは、計算条件の設定ではありませんが、便宜上ここで説明します。

通常、RYUDOでは、各原子の座標や速度の履歴は、ファイル容量節約のため、バイナリ形式で出力されています。しかし、このままでは解析に不便ですので、テキスト形式でも出力する機能を実装しています。（ファイルサイズに注意）

テキスト形式で出力したいときは、def.rdのfiles行の直後に、text行を記述します。この行はfiles行の従属行となりますので、行頭に空白が必要です。

例

```
     text    100     100     100
```

パラメータは左から順に、それぞれ、tps.rd（テキスト形式座標履歴）、tvl.rd（テキスト形式速度履歴）、tfr.rd（テキスト形式被力履歴）、の出力間隔です。

各ファイルの詳細は、別章を参照してください。

注意：テキスト形式はファイルサイズが非常に大きくなるので、大規模系や長時間（多step）計算ではご注意ください。OSの制限から、あまりに大きなファイルは出力できなくなる可能性があります。ファイルが出力できなくなると、RYUDOは異常終了します。例えば、32bitのLinuxでは、2GB以上のファイルが（追加モードで）出力できないようです。64bitのLinuxでは2GBの制限は無いようです。

7．12．外部解析プログラム呼び出し機能（Ver.1.10以降、最新Ver.1.18h）

これは、計算条件の設定ではありませんが、便宜上ここで説明します。
大規模系の長時間計算となると、ファイルサイズも膨大で解析作業が困難です。特に、より精密な解析のために tps.rd などのテキスト形式を出力する場合、そのファイルサイズは pos.rd などのバイナリ形式の10倍近くになります。そこで計算が終了してからその詳細出力を解析するのではなく、例えば RYUDO の中で計算されている MSD のように、計算の進行中に解析処理を行い、その結果だけを保存するようにすれば、大規模系の長時間計算でもファイルサイズを節約できるため、1ステップごとの解析も可能になります。
従来、このような解析機能は RYUDO のソースコードに組み込むしか方法がありませんでしたが、これは非常に詳細な制御が可能である代わりに、RYUDO のソースコードを理解しなければならないという難点がありました。そこで、RYUDO 実行中に任意の計算 step 間隔で、ユーザーが作成した解析プログラムを呼び出す機能を実装しました（呼び出し前に解析用ファイル xtr.rd も出力します）。

この機能を利用するには、def.rd の適当な場所（atom 部の前を推奨）に、以下のような行を記述してください。

例　　`extern 10 abc -n0`

ここで extern の前には空白を入れないでください。パラメータ第一項は呼び出すステップ間隔（この例では 10 step 間隔）、第2項以降は呼び出すプログラム名（この例では abc と言う名前）です。必要ならばプログラム名に引数をつけることも出来ます（この例では-n0。複数も可能）。なお、プログラム名以降は全て引数とみなされるため、この行内にコメントを記述することはできません（コロン「：」を使用しても駄目です）。
ここで出力される解析用ファイル xtr.rd は、set.rd に準じる書式となります。ただし、各原子の記述（part 部の各行）に受けている力が x,y,z 方向別にニュートン単位で追加されます。その他詳細は別章を参照してください。

ユーザーが作成する解析プログラムの開発指針としては、カレントディレクトリに専用の中間ファイルを出力して解析の進行状況を制御する形式を考えます。例として、毎ステップの各原子ごとの差分を累積するプログラムは、おおむね下記のようなアルゴリズムで構築できます。

1：起動後、まず中間ファイルが存在するかどうかを、fopen で開こうとしてチェックする。

2：中間ファイルが存在しない場合（＝初回）、set.rd と xtr.rd の各原子座標から差分を計算して、中間ファイルに出力し、xtr.rd を例えば xtr.pre という名前にリネームする（次回、上書きされるのを避けるため）。

3：中間ファイルが存在した場合、これを読み込む。現在の xtr.rd と前回リネームしておいた xtr.pre とで現在の座標差分を計算する。中間ファイルのデータに現在の座標差分を加算し、改めて中間ファイルに記録する。前回の xtr.pre を削除した上で、現在の xtr.rd を xtr.pre にリネームしておく。

以上により、RYUDO 計算終了時には、上述の中間ファイルがそのまま最終的な累積結果として利用できることになります。
同様に速度についても set.rd や xtr.rd に出力されていますので、上述のアルゴリズムで累積計算等が可能になります。（力については set.rd に記述がありませんが、これは原理的に 0step 目は RYUDO で計算されていない※1 ためです。1step 目以降で集計してください。）

※1：0step 目は set.rd を作成したプログラムによります。rdset で set.rd を作成した場合、速度は乱数で決定されており、力は算出されていません。以前の RYUDO の計算結果 end.rd から変換して set.rd を作成した場合、その RYUDO 計算の最終結果の力が 0step 目の力に該当します。

注：xtr.rd 以外のファイルを解析するプログラムをこの機能で呼び出しても構いません。その場合、呼び出しステップ間隔を、その解析したいファイルの出力間隔に合わせてください。
注：Ver.1.12b 以前では xtr.rd を出力しません。その代わり end.rd に各原子が受けた力を出力していました。ただ、この方法だと end.rd の各行が冗長になり、自作および他作の解析プログラムに影響が出たことから、別途

xtr.rd に出力し、end.rd は Ver.1.10 以前の形式に戻しました。
注：Ver.1.18g 以前では、プログラム名以降にタブコードがあった場合、自動的にスペースコードに変換していましたが、1.18f 以降はそのままプログラム呼び出しに使用します。

7．13．元素毎各種平均値出力機能(Ver.1.11 以降)

これは、計算条件の設定ではありませんが、便宜上ここで説明します。
RYUDO はデフォルトで val.rd 中に各表示方法(index)ごとの温度と MSD を出力していますが、これに加えて、各 index ごとの移動量（座標差分）および被力（受けた力）を出力させることができます。
それぞれ x,y,z の各成分、およびベクトルの大きさ（スカラー）を選択できます。また、val.rd が出力される step の瞬間値か、前回出力した step からの累計値の、2種類を選択できます。総計 16 種類にもなりますので、これらを全て出力すると val.rd が膨大になります。そこでユーザーが指定した項目だけを出力させることができるようにしています。

本機能を利用するには、def.rd の file 行下に idval 行を記述し、出力したい項目を略号（後述）で記述してください。項目は複数を空白で区切って記述できます。

例　　idval　dts　fx

この idval 行は、file 行の従属行となりますので、行頭に空白が必要です（TAB を推奨）。file 行の直下、または他の file 行従属行の下に記述する必要があります。
idval 行に記述できる項目は下記のとおりです。

・dx , dy , dz および ds

各 index に所属する全原子の座標差分*1 の平均*2 を出力します。
val.rd が出力される step における値（瞬間値）となります*3。
dx, dy, dz はそれぞれ成分ごとに平均*2 を取ります。
正負そのままで加算しますので、重心の移動と同義になります（各原子が正負均等に拡散した場合、重心は移動しないので、値は0になります）。
また、タイムステップ(dt)で割ると、その瞬間(step)の速度の成分になります。
ds はベクトルの大きさ（スカラー）で平均*2 を取ります。
全て正値にて加算しますので、その瞬間の拡散量とみなすことができます。
また、タイムステップ(dt)で割ると、その瞬間(step)の速さの平均*2 になります。
val.rd には、行頭を「id-dx」などとして、各 index ごと空白で区切って1行で出力されます。
単位はメートル(m)です。

・dtx , dty , dtz および dts

各 index に所属する全原子の座標差分*1 の平均*2 を出力します。
前回 val.rd が出力された次の step から、現在の step までの平均値となります*4。
dtx, dty, dtz はそれぞれ成分ごとに平均*2 を取り累計します。
重心の移動と同義になるのは上述の瞬間値の場合と同様です。
また、タイムステップ(dt)と出力 step 間隔で割ると、その期間の平均速度の成分になります。
ds はベクトルの大きさ（スカラー）で平均*2 を取り累計します。
これは拡散係数の算出に利用できると思われます。
また、タイムステップ(dt)と出力 step 間隔で割ると、その期間の平均速さとなります。
val.rd には、行頭を「id-dtx」などとして、各 index ごと空白で区切って1行で出力されます。
単位はメートル(m)です。

・fx , fy , fz および fs

各indexに所属する全原子の被力の平均*2を出力します。
val.rdが出力されるstepにおける値（瞬間値）となります*3。
fx, fy, fz はそれぞれ成分ごとに平均*2を取ります。
正負そのままで加算しますので、各原子が正負均等に力を受けた場合、値は0になります。
fs はベクトルの大きさ（スカラー）で平均*2を取ります。
val.rdには、行頭を「id-fx」などとして、各indexごと空白で区切って1行で出力されます。
単位はニュートン(N)です。

・ftx , fty , ftz および fts
各indexに所属する全原子の被力の平均*2を出力します。
前回val.rdが出力された次のstepから、現在のstepまでの平均値となります*4。
ftx, fty, ftz はそれぞれ成分ごとに平均*2を取り累計します。
fts はベクトルの大きさ（スカラー）で平均*2を取り累計します。
val.rdには、行頭を「id-ftx」などとして、各indexごと空白で区切って1行で出力されます。
単位はニュートン(N)です。

*1：座標差分とは、ある原子について、現在の座標と、前stepの座標との、差を取ったものです。よって、タイムステップ(dt)で割ることで、その原子の速度を求めることができます。
*2:ここでいう平均とは原子数平均のことです。例えばそのindexには100原子あるとすると、その100個の値を総和して100で割ることです。
*3:例えばval.rdを500step間隔で出力している場合、500の整数倍step（500step, 1000step, 1500step,・・・）における値になります。それ以外の（中間の）stepの値は無視されます。
*4:例えばval.rdを500step間隔で出力している場合、各500step間（1step～500step, 501step～1000step, 1001step～1500step,・・・）を通して積算した値を500で割った値になります。

7．14．外注計算機能(Ver.1.18以降、暫定公開機能)

RYUDOでは計算実行中、定期的にユーザー指定の外部プログラムを実行し（呼び出し）、そこから出力された各原子の被力データを使って（あるいはRYUDO内で計算された被力データと混合して）、計算を進めることができます。これを外注（アウトソーシング）計算と呼びます。
呼び出すプログラムとして、たとえば量子化学計算プログラム（や、その実行するスクリプト）を指定すれば、分子動力学法と量子化学計算のハイブリッド計算が可能になります。ただし、後述するように、RYUDO側のインターフェイスは最小限ですので、適宜ユーザーが変換プログラムやスクリプトを作成して、間をうまく取り持つ必要があります。

この機能を使用するには、def.rdの適当な箇所（atom部の直前を推奨）に、outsrc行を記述してください。この行のパラメーターとして、RYUDOが外注計算用に呼び出すプログラム（またはスクリプト）をコマンドライン形式で記述してください（空白可）。

例　`outsrc  qcalc -s 200 -l /opt/local/prelib/lib22.dat`

outsrc行に記述したコマンドラインは、end.rdが出力された次のstepで実行されます（すなわち外注プログラムの起動間隔はend.rdの出力間隔に一致します）。これにより起動されるプログラム（もしくはスクリプト）は、end.rdから現在の各原子の座標などを読み取り、それを元に独自の計算を行い、その結果得られた力のデータをosr.rdというファイル名で出力し、終了してください。RYUDOは、このosr.rdファイルを読み込んで、RYUDO内部で計算した力の結果と混合し、次stepの速度と座標を計算します。
なお、1step目でもoutsrc行に記述したコマンドラインが実行されますが、このときはまだend.rdが出力されて

いないので、呼び出されたプログラム（ないしはスクリプト）は、代わりに set.rd を読み込んでください。

osr.rd ファイルはテキスト形式で、1原子を1行で記述します。全原子分の行が必要です。1行は4項目からなり、左から順に、混合比率、x方向の力、y方向の力、z方向の力、となります。各項目は必ず空白文字（スペースやタブ）で間を開けてください。行頭の空白文字はあってもなくても構いません。なお、1行の文字数は最大 254 です（それ以上長い行は動作不良を引き起こします）。

例　　0.233　1.9872e-8　-2.3467e-9　7.0098e-9

混合比率は 0.0～1.0 の実数値で記述してください。0.0 で RYUDO 側 100%、1.0 で外部ファイル側 100%となります。（より正確には、1e-10 より小さい場合 RYUDO のみ、0.9999999999 より大きい場合外部のみとなり、その間（1e-10～0.999999999）の時に混合計算が行われます。）

各方向の力はニュートン[N]単位で記述してください。「e」による指数表現が可能です。

osr.rd ファイルが存在しない、osr.rd ファイルの内容に問題がある（全原子分無い、混合比率が 0.0～1.0 の範囲から外れている、他）などの場合、RYUDO はエラーを出力して停止します。ご注意ください。

8. RYUDO の実行

RYUDO プログラムの起動方法は、各 OS 環境および RYUDO のインストール状況に依存します。

Linux の場合、実行形式を PATH の通った場所においてあるならば、コマンドラインからプログラム名を打ち込むだけで起動できます。入力ファイルはカレントディレクトリにおいてあるものが使用され、計算結果はカレントディレクトリに規定のファイル名で出力されます。

カレントディレクトリに以前の計算結果ファイルが残っている場合は、上書きされますのでご注意ください。なお、不慮の事故を防ぐため、cnd.rd があるディレクトリでは RYUDO は起動しないようになっています。上書きしても良い場合は、−f オプションをつけるか、あるいはあらかじめ cnd.rd を削除しておいてください。

注：Ver.1.13 以前は、cnd.rd ではなく end.rd をチェックしていました。cnd.rd チェックは Ver.1.13a 以降です。

・起動時オプション

RYUDO には下記のオプションが指定できます。（通常の計算であれば、特に要ありません）

−c	コンティニューモードになります。
−s	画面表示の単位を SI 系にします。
−m	画面表示の単位を MXDORTO 型にします。
−f	強制起動（上述の cnd.rd のチェックを回避）します。

コンティニューモードでは、set.rd の代わりに、end.rd の座標と速度を初期値にして計算を開始します。これにより、不慮の事故で中断した長い計算を再開させたり、ステップ数が不足だった結果の追加計算が、より容易になります。（再開の場合、単に ryudo −c と起動するだけでよいです。追加の場合は、まず def.rd のステップ数を増やしてから、ryudo −c と起動してください。）

オプションの代わりに 50 以上の整数値を指定すると、RYUDO は画面にいっさいの表示をしなくなります。これはバックグラウンドで RYUDO を動かしたい場合などに有効です。しかし、画面表示が全くないのは不便ですので、代わりに scr.rd というファイルを作成し、画面に表示されるはずだった情報を書き込みます。ただし、画面表示をそのまま記録し続けるとファイルサイズが膨大になりますので、上記指定の整数値を上限として、古い行を削除するようにしています。

・計算結果のファイル出力

RYUDO の実行によって、以下のファイルが出力されます。

cnd.rd	計算の進行状況を記録しています。
end.rd	現在（最終）の系の状態（各原子の座標と速度）を記録しています。
val.rd	温度や圧力などの変化の履歴を記録しています。
pos.rd	各原子の座標変化の履歴を記録しています。
pos2.rd	pos.rd の精度向上用追加ファイルです(Ver.1.20 以降)。
vel.rd	各原子の速度変化の履歴を記録しています。
vel2.rd	vel.rd の精度向上用追加ファイルです(Ver.1.20 以降)。
for.rd	各原子にかかる力の履歴を記録しています。
for2.rd	for.rd の精度向上用追加ファイルです(Ver.1.20 以降)。
tmp.rd	各原子の運動エネルギー（温度）変化の履歴を記録しています。
log.rd	結合情報ファイルを読み込んだ場合に、読み込み状況を記録します。
scr.rd	上述の画面表示ファイルです。
tps.rd	各原子の座標変化をテキスト形式で記録しています。

tvl.rd　各原子の速度変化をテキスト形式で記録しています。
tfr.rd　各原子にかかる力の履歴をテキスト形式で記録しています。
thc.rd　熱伝導の計算結果を記録しています。
flg.rd　各原子における各種フラグの履歴をバイナリ形式で記録しています。
axs.rd　pos.rd と同じ出力間隔で各軸長の履歴をテキスト形式で出力しています。
xtr.rd　ユーザー解析用ファイル。外部解析プログラム呼び出し機能 で出力されます。

これらは設定によって出力されないものもあります。各ファイルの詳細は次章を参照してください。

・計算中の画面表示

RYUDO 実行中は以下のような情報が画面に表示されます。(表示単位に MXDORTO 型を指定した場合。)

```
Title : amo-Si surface at 300K : End=  1000
Now   0.200 sec/step. Estimated finish at  6/ 7,13:55:55 ( 2m22s later)
       MSD[A^2]      Si(  0.026)      Mo(  0.018)
  step     T[K]  P[GPa]     U[kJ/mol] density axis(x,y,z)[A]          Dy,hr:mn:sc
    41    787.7  -0.600    -162617.60   0.000  26.200  23.560  70.000  7,13:52:43
    42    782.6  -1.388    -162638.80   0.000  26.200  23.560  70.000  7,13:52:43
    43    796.5  -1.679    -162547.39   0.000  26.200  23.560  70.000  7,13:52:43
    44    835.5  -1.512    -162366.09   0.000  26.200  23.560  70.000  7,13:52:43
    45    857.4  -0.955    -162582.94   0.000  26.200  23.560  70.000  7,13:52:44
    46    836.6   0.087    -162726.93   0.000  26.200  23.560  70.000  7,13:52:44
    47    827.5   0.077    -162656.99   0.000  26.200  23.560  70.000  7,13:52:44
    48    808.4  -0.356    -162731.38   0.000  26.200  23.560  70.000  7,13:52:44
    49    791.0  -0.788    -162651.69   0.000  26.200  23.560  70.000  7,13:52:44
    50    818.6  -1.311    -162437.28   0.000  26.200  23.560  70.000  7,13:52:44
Title : amo-Si surface at 300K : End=  1000
Now   0.100 sec/step. Estimated finish at  6/ 7,13:54:19 ( 2m19s later)
       MSD[A^2]      Si(  0.031)      Mo(  0.017)
 End=  1000,Total=    865,Defect=      0,Atom= 2,Nature= 2,Index= 2
 PressX     -0.11821,  PressY     -2.32064,  PressZ     -1.49444 [GPa]
 EnergyK     8830.70,  C          0.00,  S   -171267.99,  total  -162437.28 [kJ/mol]
 Index Name        Number Temparture[K] MSD[A^2]
 Si                   864     814.593596   0.030580
 Mo                     1     462.248781   0.017493
  step     T[K]  P[GPa]     U[kJ/mol] density axis(x,y,z)[A]          Dy,hr:mn:sc
    51    838.2  -2.008    -162504.18   0.000  26.200  23.560  70.000  7,13:52:45
    52    817.2  -1.912    -162810.00   0.000  26.200  23.560  70.000  7,13:52:45
    53    769.9  -1.440    -162801.61   0.000  26.200  23.560  70.000  7,13:52:45
    54    798.7  -0.316    -162362.94   0.000  26.200  23.560  70.000  7,13:52:45
    55    823.7   0.109    -162543.39   0.000  26.200  23.560  70.000  7,13:52:45
    56    793.7   0.182    -162767.59   0.000  26.200  23.560  70.000  7,13:52:45
```

ここで黒色の部分が1行表示、緑色の部分が中間情報、青色の部分が詳細表示です(実際の画面では全て同一色です)。それぞれ表示間隔は def.rd の disp 行の第1項、第2項、および第3項で指定します。

1行表示では、現在のステップ、温度、圧力、全エネルギー、密度、セル軸長、現在の似付けと時刻、が表示されます。密度の単位は[g/cm^3]です。

中間情報では、計算のタイトル、終了ステップ数、1ステップにかかった実時間、計算終了予想日時と残り時間、各表示方法(index)ごとの MSD(平均二乗変位)が表示されます。

詳細表示では、計算終了ステップ数、原子数、欠陥数、元素種類数、性質数、表示 方法数、xyz各方向の圧力、運動エネルギー、Ewald 法によるエネルギー、その他の短距離相互作用エネルギー、左記3エネルギーの和、各表示方法(index)ごとの原子数・温度・MSD が表示されます。

このほか、disp 行の第4項で間隔を指定すれば、現在の系内の原子配置がキャラクタグラフィックスで表示されます。

注:上述の表示方法(index)ごとの MSD は、中間情報の出力時か val.rd の出力時にのみ計算されます。もし、そのいずれよりも早く詳細表示が実行された場合は、MSD の値が 0 と表示されます。また、表示方法(index)ごとの温度については val.rd の出力時にのみ計算されますので、やはりこれより先に詳細表示が行われると温度が 0 と表示されます。同様に MSD や温度の値の更新も、中間情報の出力と val.rd の出力に依存します。詳細情報

を見る際はご注意ください。
注：上述のエネルギー表示については下記のとおり注意が必要です。

- 1行表示のエネルギーは Ewald 法と水平ポテンシャル、垂直ポテンシャル、2体ポテンシャル、運動エネルギーを加算したものになります。3体および4体ポテンシャルは含まれていません。
- 詳細表示のエネルギーはそれぞれ、運動エネルギー、Ewald 法によるエネルギー、2体ポテンシャル＋水平ポテンシャル＋垂直ポテンシャル、及びそれらを加算したもの、になります。3体及び4体ポテンシャルのエネルギーは出力されません。
- 出力単位系(def.rd の disp 行で指定)に mx を指定した場合は、いずれのエネルギー表示も、第一元素(atom 部の 1)の原子数(欠陥以外)で除算さたものが出力されます。ただし、第1元素が0個の場合は除算しません。
- 水平ポテンシャルおよび垂直ポテンシャルは特殊計算機能(第7章)による1体ポテンシャルです。詳細は7．8節および7.9節を参照してください。

注：Ver.1.22d 以前では、圧力制御しない場合に密度が表示されません(0.000 になります)。

・計算中の異常表示について
RYUDO は各種設定の自由度が高い分、構造やパラメータの不備などで、計算が不安定になったり、破綻して異常終了することがなることがしばしばあります。
「Error : Too high temperature」の表示は、系内の温度が高くなりすぎて、1ステップで原子が移動しすぎる状態になった際に出力されます。一旦こうなると、計算の精度が著しく失われますので、自動的に強制終了となります。このような状態に陥る原因としては下記の問題が考えられます。
・初期構造に問題がある(近すぎる原子間がある)←計算開始後すぐ止まる場合
・ポテンシャルのエネルギー値が大きすぎる or カーブが鋭すぎる
・反発力が定義されていない or 十分でないために、原子同士が異常接近している
・設定圧力が高すぎる or セルサイズを無理に固定している
・原子の運動の激しさに対し、タイムステップが長すぎる
・その他(ケースバイケースで一概には言えません)
「Too short」の表示は、原子間距離が異常に小さくなった(通常 0.1Å 以下)場合に表示されます。計算開始後すぐに出る場合は、初期構造に問題があります(重なっている原子がある)。また、計算がしばらく進んでからこの表示が出る場合は、原子間に適切な反発力がかかっていない事が原因と考えられます。Too short の後ろには、原因となった原子番号(元素番号)が表示されますので、問題解決の参考にしてください。
「Warning : nn table over flow!」の表示は、近接原子テーブルがあふれたことを示します。この状態では、3体以上の非結合性ポテンシャルは正しく計算されていません。3体以上の cutoff 距離(cotoff 行第2項)を短くするか、テーブルの大きさ(cutoff 行第3項)を増やしてください。

・連続計算・バッチ処理について
RYUDO には単体で、複数の計算を連続して自動的に実行する機能があります。
一口に複数計算といっても、様々な概念がありますが、RYUDO では下記の2種類を実施できます。
・同一の計算モデルで、温度や圧力などの条件を順次変更しながら続ける計算
・モデルや条件など、それぞれ独立した計算を順番に実施する：複数の計算に関連性無し
RYUDO では、上記を「連続計算」、下記を「バッチ処理」と呼びます。

連続計算は、下記のようなシミュレーションを実施したい場合に利用できます。
・初期構造を一定時間常温で緩和した後、徐々に温度を上げて構造を溶かし、さらに急冷してアモ

ルファス状態を作り出す

・有機分子の流体を固体表面で徐々に挟み込む計算を行い、しばらく一定の圧力で緩和、その後固体表面を水平方向に運動させ剪断場の計算を行う

連続計算を行うには、1つの def.rd の中に、「ryudo-def」行から「run」行までを複数回記述します。例えば、上記のアモルファス構造を作る計算の場合、緩和部分、昇温部分、急冷部分、の3つのパートを記述します。

このとき、それぞれの計算におけるステップ数は連続していることに注意してください。例えば、上記3パートをそれぞれ 5000 ステップで行うとすると、最初のパートこそ「step　5000」と記述しますが、次のパートはそれに 5000 を加えて「step 10000」、さらに次のパートは「step 15000」と記述する必要があります。(もし、それぞれに「step　5000」と書いてしまうと、2番目以降のパートは「計算済み」と判断されて実施されません。)

新しいパート(2番目以降＝開始時ステップが1ではない)に移る際、各原子のフラグがクリアされます(具体的には、固定フラグ(速度無しフラグ)、温度制御対象外フラグ、運動量補正対象外フラグ、熱伝導高温側および低温側フラグ)。必要ならば新パートの atom 部で再度指定してください。また、set.rd で各原子ごと個別に指定したフラグも同様にクリアされますのでご注意ください(これは再設定方法がありません)。なお欠陥フラグ(およびそれに付随する温度制御対象外フラグ、運動量補正対象外フラグ、速度無しフラグ)は、次パートにそのまま引き継がれます。New-RYUDO 用の特殊フラグにも、引き継がれるものがあります(New-RYUDO 用追加マニュアルを参照)。

2番目以降のパートでは nature 部以降の記述が省略可能です。省略した場合、前パートの設定を引き継ぎます。ポテンシャルのパラメーターや元素の組み合わせを変更したい場合、その部だけを記述すればよいです。例えば、3体ポテンシャルのパラメーターを変更したい場合は、trio 部だけを記述し、他の nature 部、pair 部などは省略できます。ただし、部内の省略はできませんので、パラメーターなどに変更が無い組み合わせも含めて、全て前パートと同じように記述してください。

なお、全パートを通して同一のモデルを用いているため、set.rd は最初のパート用を用意するだけです(第2パート以降は、前パートの計算結果を引き継ぎます)。逆に、計算途中でモデルを変更することはできません(そもそも連続計算の意味がありません)。

注:Ver.1.11c 以前では、連続計算時に atom 部の各原子のフラグ(固定など)の処理に不具合がありました。これまでは、デフォルトでフラグが引き継がれ、フラグを外したい場合は atom 部をそのように記述する、と言うことにしていましたが、内部処理の問題でこれがうまく機能していませんでした。Ver.1.11d 以降では、基本的にフラグは引き継がれないものとし、引き継ぎたいフラグは再度 atom 部で記述すると言う方式に改めました(上記)。旧バージョンとは動作が異なることになりますのでご注意ください。

一方、バッチ処理は、下記のような場合に利用できます。

・計算条件(モデルや温度・圧力など)を変えた計算をいくつも走らせて違いを見たい

・週末など、多数の計算を入れておきたい

このためには、各個の計算用にディレクトリを作成し、その中にそれぞれの def.rd と set.rd(と必要に応じてその他ファイル)を入れておく必要があります。

注:通常、それぞれのディレクトリで複数の RYUDO を同時に走らせても、同様の結果を得られますが、CPU 数以上の計算を投入することは、処理速度の著しい低下を招きますので、バッチ処理にした方が効率が良くなります。(マルチタスクに頼って 1CPU あたりに 2 個の計算を入れた場合、ジョブ切換などのオーバーヘッドが発生しますので、それぞれの計算速度は1個の時の半分よりもさらに遅くなります。同時に走らせる計算が増えるほどこの傾向は強くなります。よって、2個以上の計算を実行したい場合は、バッチ処理にした方が、計算終了までの時間は短くなります。)

RYUDO でバッチ処理を行うには、各計算用にディレクトリとファイルを用意した後、さらに別のディレクトリ(通常は親ディレクトリ)に bat.rd という名前のファイルを用意します。このファイルの内容は、各計算ディレクトリを羅列しただけのものです(1行1ディレクトリ)。

bat.rd を置いたディレクトリで RYUDO を実行すると、RYUDO は bat.rd に書かれたディレクトリに順番に移動して、それぞれ計算を実行します。計算の進行状況は bat.rd に直接書き込まれます。

計算が終了したディレクトリは行頭に (completed) と挿入されます。
入力ファイルのエラーなどで実行できなかったディレクトリには行頭に (skipped) と挿入されます。
計算中のディレクトリには行頭に (running) と挿入されます。終了すると (completed) になります。
まだ計算していないディレクトリの行は、ユーザーが記述したそのままです。

また、バッチ処理中にマシンが落ちるなどして異常終了してしまった場合でも、通常の計算と同様 ryudo –c で続きを計算できます。もし –c を付けずに ryudo を起動した場合、(running) だった計算は再開できず、スキップされます。

参考：2CPU マシンであっても、通常はメモリやハードディスクが1つしかないので、計算を2個入れると1個だけ入れたときよりもそれぞれの速度は若干遅くなります。よって、2CPU マシンが複数台ある場合、1台に2計算を投入するよりも、2台に1計算ずつ投入する方が効率が良いです。

参考：Intel 社製 CPU のハイパースレッディング(HT)は、Linux でうまく処理できないことがあります。例えば実体は2コアまたは 2CPU で HT が掛かっている場合、Linux は全く同等の CPU が4個あると見なすため、もしここに計算を2つ投入すると、どちらか一方の CPU に負荷が集中してしまうことがあります。システム系プログラムとのジョブ切換が発生しますので、常時集中するわけではありませんが、計算速度は HT が無い場合よりも明らかに低下しますのでご注意ください。そのような場合は BIOS で HT を OFF にしておくことをお勧めします。

注：バッチ処理の仕様は Ver.1.22 で変更されました。Ver.1.21 以前では、bat.rd の最終行に「end end」と言う行が必要で、計算進行状況は bat.rd ではなく、cnd.rd に記録されます。そのため、cnd.rd および log.rd は角形さんディレクトリ個別ではなく、bat.rd のあるディレクトリに一つにまとめて（全計算分をつなげて）出力されます。

9. 出力ファイルの詳細

RYUDO で出力されるファイルの詳細を説明します。

・cnd.rd:計算進行状況出力ファイル

RYUDO を実行すると必ず出力されるファイルです。

基本的に画面に表示される情報のサブセットになります。大別して、set.rd の読み取り状況、def.rd の読み取り状況、計算の進行履歴、の3種類が記録されます。

進行履歴は、画面表示の1行表示と中間情報の MSD 表示を組み合わせたものになります。1行表示が行われる step において val.rd の出力間隔と一致する時に1行表示と同内容を記録し、中間情報が表示される step において val.rd 出力間隔の10倍の step と一致するときに MSD を記録します(val.rd の出力間隔とこれらの画面表示間隔とがうまく一致しないときは、何も表示されないことになります)。

このファイルは、各入力ファイルの読み込み状況、計算進行状況のチェックに利用してください。また、RYUDO がうまく動かない場合も、まずはこのファイルのエラー表示をチェックしてください。

注:古い RYUDO では con.rd という名前で出力していました。このファイル名は MS-Windows(古くは MS-DOS)に転送した際にトラブルとなることがあったため、現在の名前に変更されました。

・end.rd:現在状態出力ファイル

計算中に現在の各原子の座標や速度などを出力するファイルです。計算終了後は、最終状態を示すことになります(計算ステップが出力間隔で割り切れない場合でも、必ず最終状態が出力されます)。

ファイルの書式は set.rd に準じます。ただし、以下の点が異なります。

・現在の計算ステップ数が「step」行として記録されます。

・各原子の末尾に初期座標(x、y、z)が追加されます。

・各原子のフラグについて、下記の種類が追加されます。複数の記述もあり得ます。

x	固定原子
a	出現原子
l	低温原子
h	高温原子
p	部分圧力原子
s	部分温度原子

・ファイルの末尾に現在の日時が「date」行として記録されます。

このファイルの出力間隔は、def.rd の files 行第1パラメータで指定します。出力間隔に 0 を指定すると、このファイルは出力されなくなります。このファイルは毎回上書きで出力されますので、出力間隔を小さくして出力回数を増やしても、ファイルサイズは大きくなりません。ただし、出力処理には若干ながら時間を消費しますので、毎ステップ出力するのは実用的では無いかもしれません。

このファイルを用いることで、計算を容易に再開させることができます。たとえば、

・不慮の中断があった場合、最初から計算し直さずにすみます。

・計算終了後、ステップ数が足りなかったと判明した場合、容易に追加計算できます。

中断した計算を再開する場合は、単に ryudo -c と起動するだけでよいです。また、追加計算する場合も、def.rd の計算ステップ数(step 行第1項)を増やしてから、ryudo -c とするだけです。

なお、単位は全て set.rd と同様です。(各原子の初期座標はやはり無次元です。)

注:Ver.1.23 以前とは座標の出力方法が異なります。

・計算中に周期境界を越えた場合

Ver.1.23 以前　　現在座標は 0.0～1.0 の範囲内に折り返された値が出力されます。
初期座標は、現在座標が折り返された分だけ、±1.0 シフトします。
（「現在座標－初期座標＝移動量」となるように）

Ver.1.24 以降　　現在座標は 0.0～1.0 の範囲を越えて出力されます。
初期座標は計算開始時の座標をそのまま出力します。

・set.rd の座標が 0.0～1.0 の範囲外にあった場合

Ver.1.23 以前　　0.0～1.0 の範囲に折り返して計算を開始します。
end.rd に出力される初期座標も、この折り返された値になります。

Ver.1.24 以降　　0.0～1.0 の範囲に折り返して計算を開始しますが、
end.rd に出力される初期座標は set.rd そのままです。

end.rd を参照するツール類をご使用の際はご注意ください。

・val.rd：統計的データ出力ファイル

温度や圧力など、系内の統計的データを記録していくファイルです。以下に例を示します。

```
axis     2.62000000e-09  2.35600000e-09  7.00000000e-09 90.0 90.0 90.0
temper      691.8531408     682.6863706       0.0000000
press   -8.09419302e+08 -7.92745426e+08 -8.40264746e+08 -7.95247735e+08 0 0 0
energy   0.00000000e+00  0.00000000e+00 -2.08610142e-16  1.23938831e-17
density  0.00000000e+00
id-temp  6.8082996606e+02  2.2866198585e+03
id-msd   5.5139805481e-21  1.2047791783e-20
end
```

axis 行は、現在のセルサイズ[m]、およびセル角度[度]を示します。（現在の RYUDO では、角度は 90 度固定ですが、将来の拡張のため、出力するようにしています。）

temper 行は、現在の瞬間温度[K]、val.rd 出力間隔での平均温度[K]、温度制御用平均温度[K]を示します。なお、温度制御をしていない場合は、第3項が 0 になります。

press 行は、現在の圧力（全圧、x方向、y方向、z方向、α 方向、β 方向、γ 方向）[Pa]を示します。（α～γ 方向は、将来の拡張用であり、現在は 0 が出力されます。）

energy 行は、現在のエネルギー（Ewald 法、3・4体ポテンシャル、2体ポテンシャル※1、運動）[J]を示します。

density 行は現在の密度[kg/m^3]を示します※5。

id-temp 行は、各表示方法(index)ごとの温度を示します。系内全原子の運動エネルギーを所属 index ごとに毎 step 積算し、val.rd の出力時に平均値（粒子数と val.rd 出力間隔 step 数で割る）として算出したものです（積算値は val.rd 出力後にクリアされます）。※2、※3

id-msd 行は、各表示方法(index)ごとの MSD を出力します。系内全原子の val.rd 出力時点での MSD を、所属 index ごとに平均した値です。※3

(id-dx 等（7.13 参照）は id-msd の下に同形式で出力されます)。

set.rd で分子番号を使用している場合（set.rd の part 部の各行で第2項に 1 以上を指定している行がある場合）、分子別に集計した MSD も同時に出力されます。mc-msd と mm-msd の2種類があります※4。

end 行は、1回分の出力の終わりを示します。

val.rd は、files 行の第2項に設定したステップ間隔ごとに、上記の情報が追加されてゆきます。間隔が短すぎると、ファイルの大きさが増えますので注意が必要です。また解析プログラムにもよりますが、pos.rd の実セル長をここから取得する場合は、pos.rd と同じ出力間隔にしたほうが安全です。

※1：2体ポテンシャルエネルギーには、特殊計算機能の水平ポテンシャルおよび垂直ポテンシャルによるエネルギーも含まれています。水平ポテンシャルおよび垂直ポテンシャルの詳細は7.8節及び7.9節を参照してください。

※2：Ver.1.10 以前では、id-temp と id-msd はそれぞれ el-temp と el-msd と言う名称でした。Ver.1.11 の元素毎各種平均値出力機能の追加にともない、el-から id-に変更しました。

※3：Ver.1.10 以前では、id-temp の平均値算出は val.rd 以外にも、中間情報および詳細表示の際にも行われ、積算値がクリアされていました。このままでは解析に混乱を生じる可能性があったため、Ver.1.11 以降、上述の仕様に変更されました。

※4：分子別 MSD 出力機能は Ver.1.21 以降に実装されています。

分子別の MSD が出力されるのは、set.rd で1以上の分子番号を使用しているときだけです。

分子番号が0しかない場合は出力されません。

分子内番号による結合情報だけは分子の識別ができないので出力されません。

逆に分子番号が全て1の場合は出力されます（すなわち全原子の MSD になります）。

出力は id-msd 行の下に続く下記の2行になります。

・mc-msd 行は座標中心＝分子に属する原子の平均座標の二乗変位です。

原子の質量を考慮していません。擬似的な体積中心と言えます。

・mm-msd 行は質量中心＝分子に属する原子の座標に原子の質量を加味して平均した二乗変位です。

いわゆる重心の MSD です。原子の質量としては def.rd の値が使われます（set.rd の値は使いません）。

単位はいずれも m^2 です。各項は左から順に、分子番号0、1、2・・・の MSD となります。もし分子番号が連続して使われていない場合、存在しない分子番号の項は出力されません。たとえば、set.rd 中の分子番号に0，1，4のみが使われていた場合、mc-msd 行および mm-msd 行はいずれも3項だけ値が出力されます（0、1、4の分）。

分子別 MSD 出力が不要の場合、コンパイル時に変更することが出来ます。defi.h の#define MOLESD の値を変更してからコンパイルしてください。デフォルトは3（＝mc-msd と mm-msd の両方を出力）です。mc-msd のみ出力したい場合は1、mm-msd のみ出力したい場合は2、両方とも止めたい場合は0に変更してください。

※5：Ver.1.22d 以前では、密度の計算は圧力制御ルーチンでのみ行っているので、圧力制御無し（＝セルサイズ一定）のときは出力されません（0.000 になります）。

・pos.rd：全原子座標履歴ファイル

すべての原子の座標を、def.rd に指定された間隔(files の第3項）で記録したファイルです。Ver.1.20 以降では精度向上のための追加ファイル pos2.rd も同時に出力されます（デフォルト）。これらはいずれもディスク容量を節約するためバイナリ形式で出力されますので、more コマンドなどで中身を見ることはできません。適宜専用の解析プログラムをご使用ください。

これらのバイナリ形式は以下の方法で出力されています。ご自分で解析プログラムを作る際などの参考にしてください。（注：RYUDO 内の座標はもともと部分座標：軸長で割った規格化値で 0.0～1.0 の範囲です）

1：全原子について、下記2～6を繰り返す

2：x 方向、y 方向、および z 方向について、下記3～6を繰り返す

3：部分座標を 10000 倍して整数(符号付き 16bit)化する＝0～9999 の整数になる→pos.rd 用

さらに pos2.rd 用に、小数点以下を 10000 倍して整数化=同じく 0～9999 の整数

4：pos.rd 用整数座標に、周期境界を越えた回数（-3～+2）※1を 10000 倍して加算

＝ -30000～29999 の整数になる

（もし周期境界を越えた回数が-3～+2 の範囲を超える場合、範囲内に折り返すと共に、

上位桁相当の6回分を 10000 として pos2.rd 用整数座標に加算※1）

5：各整数座標を上位 8bit と下位 8bit に分割（pos.rd および pos2.rd とも、以下同じ）

6：まず上位 8bit を1文字（1バイト）として出力し、次いで下位 8bit を出力

上記4：の処理により、ここで出力される座標は周期境界条件なしと同じイメージになります（すなわち、部分座標で 0.0～1.0 の範囲外、整数座標で 0～10000 の範囲外に出ることがあります）。これは pos.rd（および pos2.rd）から移動量や拡散係数、MSD の算出を可能にするためです。

※1：符号付き 16bit 整数の制限を考慮し、周期境界を越えた回数は-3～+2 の範囲でしか加算しません。それ以上の回数は pos2.rd の 10000 の位を上位桁として表現します（pos2.rd の 10000 の位を周期境界越え6回分とします）。たとえば、正方向に 3 回越えた場合、pos2.rd に6回分として 10000 を加算しつつ、pos.rd には-30000 を加算します。これ

により、1x6 − 3 = +3 回を表現できます。もし pos.rd だけを見て視覚化した場合、整数座標で+30000 を越えると反対側の−30000 の位置に出ることになります（すなわち6倍のセルサイズの中でループしたようになります）。なお、pos.rd と pos2.rd と合わせた場合でも、−21〜+14 回を超える周期境界越えは表現できません。もし超えた場合は、それぞれ最小値・最大値の反対側に出るように表現します（すなわち 36 倍のセルサイズの中でループしたようになります）。−21〜+14 回を超えるような計算で移動量などを解析したい場合は、テキスト形式出力をご利用ください。

※2：Ver.1.19 以前の RYUDO では pos2.rd を出力しません。また、周期境界を超えた回数が−3〜+2 に収まらなかった場合の処理が異なります。整数化座標で言うと、−30000 より小さくなった場合（周期境界を負方向に4回以上越えた場合）は−20000 に引き戻されます。また、+30000 よりも大きくなった場合（周期境界を負方向に3回以上越えた場合）は+20000 に引き戻されます。

pos.rd および pos2.rd は、def.rd に指定された step 間隔ごとに追加出力されます。あまり頻繁に出力させるとファイルサイズが大きくなりすぎますのでご注意ください（ハードディスク容量を圧迫したり、解析プログラムで読み込めなくなったりします）。

・vel.rd：全原子速度ファイル

すべての原子の速度を、def.rd に指定された間隔(files の第4項)で記録したファイルです。Ver.1.20 以降では精度向上のための追加ファイル vel2.rd も同時に出力されます（デフォルト）。これらは pos.rd（および pos2.rd）と同様に、バイナリ形式で記録されていますので、解析には別途プログラムが必要です。

バイナリ形式の出力は pos.rd および pos2.rd と同様に下記の方法で行っています。（注：RYUDO 内の速度は[m/s]単位になっています。）

1：全原子について、下記2〜5を繰り返す
2：x 方向、y 方向、および z 方向について、下記3〜5を繰り返す
3：原子速度を[m/s]のまま符号付き 16bit で整数化し、vel.rd 用とする（−32768〜+32767m/s）
　さらに vel2.rd 用に、小数点以下を 10000 倍して 0〜9999 の範囲の 16bit 整数にする
4：上記 16bit 整数を、上位 8bit と下位 8bit に分割（vel.rd、vel2.rd とも、以下同じ）
5：まず上位 8bit を1文字（1バイト）として出力、つづいて下位 8bit を出力

（vel2.rd の 10000 の桁（およそ 2bit 強の情報量）は今のところ未使用です。）

vel.rd および vel2.rd は、def.rd に指定された step 間隔ごとに追加出力されますので、あまり頻繁に出力させるとファイルサイズが大きくなりすぎますのでご注意ください。

・for.rd：全原子被力ファイル

すべての原子がそれぞれ受けている力を、def.rd に指定された間隔(files の第5項)で記録したファイルです。Ver.1.20 以降では精度向上のための追加ファイル for2.rd も同時に出力されます（デフォルト）。これらは pos.rd（および pos2.rd）と同様に、バイナリ形式で記録されていますので、解析には別途プログラムが必要です。

バイナリ形式の出力は pos.rd および pos2.rd と同様に下記の方法で行っています。（注：RYUDO 内の力は[N]単位になっています。）

1：全原子について、下記2〜5を繰り返す
2：x 方向、y 方向、および z 方向について、下記3〜5を繰り返す
3：力を[pN]に換算してから符号付き 16bit で整数化し、for.rd 用とする(−32768〜+32767pN)
　さらに for2.rd 用に、小数点以下を 10000 倍して 0〜9999 の範囲の 16bit 整数にする
4：上記 16bit 整数を、上位 8bit と下位 8bit に分割（for.rd、for2.rd とも、以下同じ）
5：まず上位 8bit を1文字（1バイト）として出力、つづいて下位 8bit を出力

（for2.rd の 10000 の桁（およそ 2bit 強の情報量）は今のところ未使用です。）

for.rd および for2.rd は、def.rd に指定された step 間隔ごとに追加出力されます。あまり頻繁に出力させるとファイルサイズが大きくなりすぎますのでご注意ください。

・tmp.rd：全原子運動エネルギーファイル
すべての原子の運動エネルギー（温度）を、
def.rd に指定された間隔(files の第5項)で記録したファイルです。
pos.rd と同様、バイナリ形式で記録されていますので、解析には別途プログラムが必要です。
バイナリ形式の出力は下記の方法で行っています。

1：各原子の運動エネルギーを温度[K]に換算
2：換算した温度をそのまま整数化(16bit)→小数点以下切り捨て
3：整数(16bit)化された数値を、上位 8bit と下位 8bit に分割
4：まず上位 8bit を1文字（1バイト）として出力、つづいて下位 8bit を出力
5：全原子について、1～4を繰り返す

tmp.rd は def.rd に指定された間隔ごとに記述が追加されていきますので、あまり頻繁に出力させますと、ハードディスク容量を圧迫したり、解析プログラムで読み込めなくなったりしますので、ご注意ください。

・tps.rd：全原子座標ファイル（テキスト形式）
pos.rd と同様、全原子の座標を逐次記録するファイルですが、pos.rd がバイナリ形式であったのに対し、tps.rd ではテキスト形式で出力されます。これにより、独自の解析プログラムなどの作成が容易になるほか、座標の精度も確保できます(pos.rd では実質4桁の精度しかありません）。一方で、ファイルサイズが非常に大きくなりますので、計算ステップ数、原子数、出力間隔に注意が必要です。
以下にファイルの内容を例示します。

```
axis     2.092267385352669e-09 2.081747525474444e-09 1.392681656459950e-09
1:O^2-    0.374282511679340        0.050091092422330        0.758921370698567
1:O^2-    0.125914771712974        0.948095661608534        0.258614294663550
1:O^2-    0.624992622099729        0.549524764418257        0.238933083162002
2:Si^2+   0.420209015130551        0.056679868129164        0.665440308805454
2:Si^2+   0.079637341568431        0.942979639945443        0.165512978234313
2:Si^2+   0.580156552893353        0.556768906929942        0.333552217372585
3:Al^3+   0.311948823415684        0.326293497674753        0.180830886822238
4:Na^+    0.496064029308648        0.496622787387862        0.002929194691235
end
```

ここで、akis の行がセルサイズ[m]を示すとともに、1ページ分の開始を表します。続いて各原子ごとに、元素No.:元素名、x座標、y座標、z座標の順に記録されます。座標はすべて部分座標です（単位無し）。最後に end の行で締めくくります。
def.rd に指定された間隔(text 行第1項）ごとに、tps.rd には上記 axis～end までが追加されます。
あまり頻繁に出力させますと、ハードディスク容量を圧迫しますのでご注意ください。

・tvl.rd：全原子速度ファイル
vel.rd と同様、全原子の速度を逐次記録するファイルですが、vel.rd がバイナリ形式であったのに対し、tvl.rd ではテキスト形式で出力されます。これにより、独自の解析プログラムなどの作成が容易になるほか、値の精度も確保できます(vel.rd では-32768～32877[m/s]まで）。一方で、ファイルサイズが非常に大きくなりますので、計算ステップ数、原子数、出力間隔に注意が必要です。
ファイルの書式としては、tps.rd から axis 行を取り除いたものに準じます。def.rd に指定された間隔(text 行第2項）ごとに、最初の原子～end 行までが繰り返し記述されます。数値の単位は[m/s]です。

・tfr.rd：全原子被力ファイル
for.rd と同様、全原子がそれぞれ受ける力の大きさを逐次記録するファイルですが、for.rd がバイナリ形式であったのに対し、tfr.rd ではテキスト形式で出力されます。これにより、独自の解析プログラムなどの作成が容易

になるほか、値の精度も確保できます(for.rd では-32768～32877[pN]まで)。一方で、ファイルサイズが非常に大きくなりますので、計算ステップ数、原子数、出力間隔に注意が必要です。
ファイルの書式としては、tps.rd から axis 行を取り除いたものに準じます。def.rd に指定された間隔(text 行第3項)ごとに、最初の原子～end 行までが繰り返し記述されます。数値の単位は[N]です。

・thc.rd：熱伝導計算ファイル
def.rd にて cond の記述を行い、熱伝導計算を実施した場合に出力されるファイルです。
以下にその例を示します。

```
Step,    Temp.,   LoTemp, HiTemp, LowEnergy[J],    HighEnergy[J],   LowEnergyS[J],   HighEnergyS[J]
     100,   370.9,   278.4,   524.4,   6.0984449e-20, -1.0929429e-19,   0.0000000e+00,   0.0000000e+00
     200,   431.4,   334.0,   455.1, -1.5301740e-19,   3.2068997e-19,   0.0000000e+00,   0.0000000e+00
     300,   439.8,   274.2,   492.1, -6.5430227e-19,   3.6493591e-20,   0.0000000e+00,   0.0000000e+00
            ・・・
    3700,   460.3,   285.5,   477.6,   8.6984698e-20,   1.4413814e-19, -2.4134119e-18,   1.8054827e-18
    3800,   319.2,   297.9,   481.6,   6.1897632e-21,   4.6184647e-20, -3.0872629e-18,   9.1692270e-19
    3900,   381.4,   268.3,   577.4,   1.5795278e-19, -2.5398951e-19, -2.8283404e-18,   1.1536329e-18
    4000,   325.6,   303.9,   451.5, -8.3655205e-21,   1.4247757e-19, -3.5672638e-18,   1.0859657e-18
```

ここで最初の行は各項目の注釈です。
各行は左から順に、ステップ数、非制御部分の温度、低温制御部分の温度、高温制御部分の温度、低温側に加えられたエネルギー、高温側に加えられたエネルギー、低温側に加えられたエネルギーの累計、高温側に加えられたエネルギーの累計、となります。温度の単位は[K]、エネルギーの単位は[J]です。
エネルギーの累計は、cond 行第2項で指定されるステップまで始まりませんので、それまでは常に 0 になります。
この例は、低温側および高温側範囲を、atom 部のフラグで指定(L/H)した場合の出力です。cond 行に伝達方向と範囲を記述した場合は、各行最後に現在の低温側原子数と高温側原子数が追加出力されます。

・log.rd：各種記録ファイル
RYUDO 実行時に、結合情報の読み取り結果、および2体ポテンシャルで組み合わせ00を指定したときの展開情報が出力されるファイルです。また、計算中にプロトン伝導や反応表現のイベントなどの不定期情報を記録するためにも使われます。

・scr.rd：画面情報記録ファイル
RYUDO を数字オプションを付けて起動したときに、画面表示に変わって作成されるファイルです。指定された数字が行数の上限となりますので、放置しても大きくなりすぎることはありません。
数字オプション付きの起動については、別章を参照してください。

・flgrd：原子フラグ履歴ファイル
RYUDO で計算中の全原子の各種フラグを記録するファイルです。バイナリ形式で出力され、1原子あたり1バイトになります。計算中の出力間隔は pos.rd と一緒になります。よってファイルサイズは、出力回数×原子数×1バイト、となります。(出力回数は、全ステップ数÷出力間隔、になります)。これはちょうど pos.rd の6分の1のサイズになります。
フラグの内容は下記の通りです。それぞれ1で ON、0で OFF です。

1ビット目：欠陥フラグ
2ビット目：固定フラグ
3ビット目：温度制御無しフラグ
4ビット目：運動量補正無しフラグ

5ビット目：高温部（熱伝導）フラグ
6ビット目：低温部（熱伝導）フラグ
7ビット目：結合フラグ（bondon.dat/bnd.rd に結合が存在する原子）
8ビット目：オプションフラグ（将来用）

なお、フラグの内容に関しては、将来変更する可能性があります。

・axs.rd:セル軸長履歴ファイル（Ver.1.09 以降）

pos.rd の視覚化および解析の補助用として出力されるファイルです。pos.rd と同じ出力間隔で、現在のセル軸長（x、y、z）が 1 行ごとに出力されます。行頭には現在の計算 step 数が出力されます。テキスト形式ですので、more コマンドで見たり、Excel に取り込むことも容易ですが、val.rd もあるので、その必要性は低いと思われます。なお、軸長の単位は[m]です。

・xtr.rd：解析用ファイル(Ver.1.13 以降)

外部解析プログラム呼び出し機能（def.rd に extern 行を記述）を使用する際に出力されるファイルです。extern 行に記述したステップ間隔で毎回上書きされます。基本的な書式は set.rd に準じます。ただし下記の点が異なります。

・cond 行の温度と圧力は、現在の系の瞬間値となります。
・現在の計算ステップを step 行として、cond 行の次に出力します(end.rd と同様)。
・各原子記述(part 部の各行)が若干変更されます。速度の単位が[m/s]になります。末尾に現在その原子が受けている力を x,y,z 方向別に追加出力します（単位はニュートン(N)）。いずれも現在の状態を示します。以上より、行内各項は左から順に、元素種、分子番号、分子内番号、現在のフラグ、現在の部分座標(x/y/z)、現在の速度(x/y/z)[m/s]、現在の被力(x/y/z)[N]、となります。
・ファイルの末尾に date 行を記述し、現在の日付を出力します（end.rd と同様）。

下記に出力例(抜粋)を示します。

```
ryudo-xtr 1.00
title     sample
axis      1.107800000000000e-08  5.000000000000000e-09  2.960000000000000e-09 90.0 90.0 90.0
cond      353.338941071137[K] -2388751390[Pa] -3583873590[Pa] 59885444[Pa] -3570441669[Pa]
step      200
elem      No.       Name      Weight[au]
          1         c1        12.0100000
          2         c2        12.0100000
          3         c3        12.0100000
part
   1   0   0   b  0.696544718316033  0.567156442248992  0.511624201476373 1.127611769590422e+03 1.
731232208714550e+02 6.871310355572089e+02 2.333598199034652e-09 -1.289848326161689e-10 7.358118422
281172e-09
   1   0   0   b  0.705417452222422  0.608266672012285  0.523230579417609 -7.084618170613594e+02 1.
136979243246304e+02 1.098169315337051e+02 -2.371887179944112e-09 -8.789512224408459e-10 -7.8142739
61653595e-10
date 04:36:19, 05/27
```

参考：バイナリ履歴出力の精度向上用追加ファイル

上述の pos2.rd、vel2.rd および for2.rd はデフォルトで出力するようになっていますが、defi.h の#define EXTPVFOUT を 0 にしてコンパイルすることで、出力させないようにすることもできます。これにより、テキスト形式出力の解析がメインの場合などにバイナリファイル出力を節約することができますが、計算後に pos2.rd 等を消しても効果は一緒なので、あまり意味はないかも知れません。

なお、pos2.rd 等の出力をしないようにすると、pos.rd の周期境界越えの記録方法も変更され、Ver.1.19 以前と同様になります（部分座標で言うと、+3.0 を超えると+2.0 に引き戻され、−3.0 を超えると−2.0 に引き戻されるようになります）。

10. 支援プログラム

RYUDOは計算本体プログラムのため、実際の利用に際しては各種支援プログラム類の利用が効果的です。標準で付属するものとして、名称がrdで始まる下記のツール類があります。

（初期ファイル準備系）

・rdset （最新Ver.1.36b）

set.rdの作成を支援するプログラムです。

ソースコードは「rdset.c」です。これをコンパイルするには、例えば下記のようにしてください。（mathライブラリを使用します。）

例　gcc rdset.c -o rdset -lm

実行時はオプションとして構造ファイルと温度を指定します。

例　rdset strcture.rst 300.0

構造ファイルとしては、RYUDOのオリジナルであるrst形式のほか、xtaldata形式、car形式、xtl形式、およびxtal.mdy形式が使用できます。

- rst形式の書式はset.rdに準じます。set.rdとの相違点は、最上行が「Structure0.31」になり、cond行は無くなり、part部の各原子の速度も無いことです。なお、elem部の各行第3項の質量は、初期速度の算出に使用されますので、必ず正しい値を入れてください。set.rdではdef.rdの設定が優先されるため、質量の記述はほとんど不要ですが、rstファイルでは大変重要なのでご注意ください。
- xtaldata形式は、TRIBOSIMやMSPORTなどで用いられている結晶データベースファイルに準じます。ただし、1ファイル1構造に限られる他、空間群はP1型のみに対応しています。
- car形式、およびxtl形式は、Cerius2プログラムで用いられている構造ファイルです。car形式は周期境界条件のあり／なし両方に対応しています。
- xtal.mdy形式は、Colorsプログラムで用いられている構造ファイルです。

rst形式以外の場合、ファイル中に各元素の重さが記述されていませんので、rdsetが対話的に訪ねてきます。それぞれ適切な質量を[au]単位で入力してください。

各ファイル種別の判断は、ファイル名で行っています。拡張子が.rstのものはrst形式、.carのものはcar形式、.xtlのものはxtl形式と判断します。また、ファイル名がxtal.mdyのものはxtal.mdy形式、以上いずれにも該当しないものはxtaldata形式とします。このため、ファイル名（拡張子）は大変重要になりますので、ご注意ください。特に、xtal.mdy形式はこのファイル名しか認めません。

温度は[K]（ケルビン）単位で指定してください。この温度に準じた初期速度が乱数で生成され、ボルツマン分布に基づいて各原子に割り当てられます。

オプションとしては、さらに「乱数の種」を指定することができます。同じ乱数の種を使えば、同じ初期速度が生成されます。

例　rdset strcture.rst 300.0 12

rdsetプログラムが終了すると、カレントディレクトリにset.rdが作成されます。もし既存のset.rdがあった場合は上書きされ、以前のset.rdは失われますのでご注意ください。

・rdend （最新Ver.1.00）

RYUDO用のend.rdファイルをset.rdファイルに変換するツールです。ただし、元来end.rdとset.rdは、ほとんど同じ書式ですので、単純変換ならばこのツールを使う意味はありません。

このツールでは以下の変換作業を行います。

- step, data などの不要行を削除する
- cond 行が 80 字未満に収まるように、圧力値の桁数を少なくする
- 各原子の記述が 80 字未満に収まるように、座標と速度の桁数を減らし、初期座標以降を削除する
- 各原子の記述(part 部)からオプションで指定されたフラグを削除する

(注:80 字未満にするのは、単に人が見やすくするためで、set.rd に必須のことではありません。)

このツールの使用意義は上記の「フラグ削除」と、構造を手作業で修正して別の計算をしたい場合にあります。上述のように、座標や速度などの桁数が減らされてしまいますので、単に set.rd に変換したいだけなら、従来通り手作業で書き換えたほうが、精度が維持できて良いでしょう。さらに、単純に step 数を増やして続きの計算したいだけならば、継続計算(ryudo -c)の方をお勧めします。

ソースコードをコンパイルする際は、例えば「 gcc rdend.c -o rdend 」などとしてください。(math ライブラリは使用していません。)

使用方法は、カレントディレクトリに end.rd を置いて、rdend を実行するだけです。もし既存の set.rd があった場合は上書きされますのでご注意ください。各原子のフラグで削除したい物がある場合は、引数として指定してください (例:x を削除する場合「 rdend x 」)。

・rdrho (最新 Ver.1.03a)

単斜晶(2軸が 90 度)セルを 斜方晶(3軸が 90 度＝直方体)セルへ自動変換するプログラムです。現在の RYUDO では、斜方晶セル(全軸 90 度)しか計算できませんので、単斜晶構造を使いたい場合は、あらかじめこのツールで変換してください。最新版は Ver.1.03 です。

ソースコードをコンパイルする際は、例えば「 gcc rdrho.c -o rdrho -lm 」などとしてください。(math ライブラリを使用しています。)

使用方法は下記のように入力ファイル名をつけて起動するだけです。

rdrho (入力ファイル名)

入力ファイルとしては rst 形式のほか、P1 型の car 形式に対応します。変換結果は rst 形式で画面に表示されますので、適宜リダイレクトしてファイルに保存してください。

現在、本ツールで変換できるのは、ある1軸が 100 度以上の単斜晶に限られます。100 度以上となるのは、α、β、γ のいずれでも構いません(自動的に判断します)。また、残る 90 度の2軸については、長さが異なっていても構いません。

入力ファイルに car 形式を使用した場合、その各原子記述行の第1項から後ろの数字部分を取り除いた部分が、出力される rst 形式の元素名になります。

対応できる元素種類数は、デフォルトのソースコードで最大32です。それ以上に対応させたい場合は、ソースコードの #define ELEMS を変更して再コンパイルしてください。また、結合情報は1原子あたり9個までです。原子数の上限はありません。

本ツールでは、90 度軸の一方を整数倍して斜方晶化します。たとえば、90 度の2軸が同じ長さの場合、残る軸が 120 度の時は2倍、109.5 度のときは3倍、104.5 度の時は4倍、・・・、として直方体化しています。これらに一致しない角度の場合、適宜近い角度に見なして変換しますので、構造に若干ズレが生じます。いずれにせよ、<u>変換後の構造は 緩和計算を十分行うなど、留意してご使用ください。</u>また、90度でない軸の角度が小さいほど、整数倍値は大きくなりますので、構造の肥大化にご注意ください。

> より正確に言うと、本ツールでは「 n = - x / (y * cos(c)) 」という式を用いて、整数倍率を決定しています。ここで、n が倍率、c が 100 度以上の角、x と y はその軸に直行する辺(＝90 度の軸)の長さです。ここで n が整数でない場合は、四捨五入して近似します。よって構造にズレが生じます。

・rdext (最新 Ver.1.03)

RYUDO 用構造データである rst ファイルを、x、y、z方向にそれぞれ任意の整数倍に拡張するツールです。構造データ中に含まれる分子構造(分子番号、分子内番号、および結合情報)を 維持したまま複製することができますので、CVFF 等の分子内の結合情報を必要とするポテンシャルを用いるモデルを多数複製する場合などに便利です。(最新は Ver.1.02)

ソースコードをコンパイルする際は、例えば「 gcc rdext.c -o rdext 」などとしてください。(math ライブラリは使用していません。)

使用方法は「 rdext (元 rst ファイル名) (x 方向倍数) (y 方向倍数) (z 方向倍数) 」となります。結果は画面に出力されますので、ファイルに保存するには適宜リダイレクトしてください。

なお、結合情報は、分子モデル(=結合に関与する各原子の分子内番号が1以上)の系にのみ対応します。全原子に通し番号を振る(=各原子の分子番号を0とし、かつ、分子内番号を通し番号とする)モデルには対応していません。また、結合が周期境界をまたがるような系にも対応していません。

・rduni　(最新 Ver.1.01)

複数の rst ファイル(RYUDO 用構造ファイル)を一つに統合するツールです。各 rst ファイルのセルサイズの違いを調整し、元素種番号を付け直して統合します。

ソースコードをコンパイルする際は、例えば「 gcc rduni.c -o rduni 」などとしてください。(math ライブラリは使用していません。)

実行するにはコマンドラインから「rduni (rst ファイル 1) (rst ファイル 2) …」のように入力してください。ファイル名は2つ以上指定します。変換結果は画面に出力されますので、適宜リダイレクトしてファイルに保存してください。

各構造のセルサイズが違う場合、一番大きいサイズに合わせられます。このとき、小さい方の構造は中心に寄ります (オリジナルの原子間距離が維持されます)。

結合情報(bond 部)も適宜統合されますが、まだ不完全な部分があります。結合情報を使用する場合は、変換後の結果をよく確認し、問題がある場合は適宜修正してご使用ください。

デフォルトのソースコードでは、最大元素種類数は 64、結合先情報は1原子当たり7つまでです。
より大きな系に対応させるには、ソースコードの定義部分(#define)を適宜修正してください。

・rdsrf

三次元周期境界条件がかかったままで疑似表面構造(薄膜+空間)を作成するための支援ツールです。まだ表面用の空間がないバルクモデルから、任意の方向に空間を空けたモデルを作成します。

ソースコードのコンパイルは、たとえば「 gcc rdsrf.c -o rdsrf 」などとしてください。(math ライブラリは不要です。)

あらかじめ、表面構造用にバルクモデルを作成し、rst 形式で保存しておいてください。セル境界が表面になりますので、適宜位置合わせしてください。

このツールを実行するには、コマンドラインから「 rdsrf (構造ファイル名) 」のように入力してください。

- ・構造ファイル名は必須です。rst 形式に準じたファイルが使用できます(set.rd や end.rd も可能です)。
- ・デフォルトでは、z軸方向に空間を空け、z 軸方向のセルサイズを 100Å にします。元のバルク構造の z 軸方向位置は、中心がz軸長の 25%になります。

結果は rst 形式で画面に出力されますので、ファイルに保存するにはリダイレクトしてください。

空間を空ける軸方向、空けた後の最終的な軸長、表面構造の中心位置、についてはオプションで指定できます。これは「 rdsrf (構造ファイル名) [方向] [軸長] [中心位置] 」のように記述します。

- ・方向は、x、y、z のいずれか英字一文字で指定します。
- ・軸長はÅ単位で指定します。
- ・中心位置は部分座標(軸長における率)で指定します(例:25%の場合は 0.25)。

各項は後ろから順に省略可能です。

（解析関係）

・rdtxt　（最新 Ver.1.03）
各種解析に便利なように、RYUDO のバイナリ形式の出力ファイルを、テキスト形式に変換するプログラムです。変換できるファイルは、pos.rd、vel.rd、for.rd、tmp.rd です。
ソースコードをコンパイルする際は、例えば「 `gcc rdtxt.c -o rdtxt` 」などとしてください。（math ライブラリは使用していません。）
使用方法としては、上記の内、変換したいファイルと、set.rd をカレントディレクトリにおき、rdtxt を起動するだけです（変換したいファイル名を指定する必要はありません。カレントディレクトリにある対象ファイルが全て変換されます）。変換結果はそれぞれ、pos.txt、vel.txt、for.txt、tmp.txt というファイル名でカレントディレクトリに保存されます（もし既存の同名ファイルがあった場合は上書きされますのでご注意）。
変換後の書式は、1原子1行として、x, y, z それぞれの方向の値が並んで出力されます（hbc.txt をのぞく）。tmp.txt の場合は方向の別がないので値は一つです。1ページ分の出力が終わった後は「end」という行が書き込まれます。値の単位は、pos.txt が無次元（部分座標）、vel.txt が[m/s]、for.txt が[pN]、tmp.txt が[K]です。
各テキスト形式の数値精度は、元となるバイナリファイルの制限により、3～5桁程度しかありません。より精度の高い解析を行うためには、計算前にあらかじめ def.rd でテキスト形式出力を指定してください（file 行の従属行で text 行を指定、詳細は別章を参照のこと）。本 rdtxt プログラムの利用は、過去の計算でバイナリ形式しかなく、かつ再計算が難しい場合に限定した方がよいでしょう。
変換後のテキストファイルは RYUGA などで視覚化したり解析したりすることができません。あくまで、ユーザーが自分で解析（独自の解析プログラムを作ったり表計算ソフトに読み込ませたり）するための形式になります。ご注意ください。

・rdsum　（最新 Ver.1.01）
RYUDO の計算結果（各原子の座標、速度、力の履歴）から、元素ごとの総和とその累計を算出するツールです。
ソースコードをコンパイルする際は、例えば「 `gcc rdsum.c -o rdsum -lm` 」などとしてください。（math ライブラリを使用しています。）
使用方法としては、カレントディレクトリに必要なファイル（後述）を配置してから、下記の様にファイル名と対象元素種番号をつけて起動するだけです。（必要に応じて解析方向(x/y/z)も付けます。）

rdsum　(filename)　(elementNo.)　[x/y/z]

ここで、(filename)が解析したいファイル名で、 pos.rd, vel.rd, for.rd, tps.rd, tvl.rd, tps.rd のいずれかを指定します。(elementNo.)が解析したい元素の種類番号(set.rd の elem 部に準拠)です。この後ろに x、y、z のいずれか1文字を指定すると、その方向成分のみを計算し、何も指定しないとベクトルの大きさを計算します。
このツールを実行する際は、解析したい履歴ファイル（上記いずれか）の他に、 set.rd も必要になります。また、knd.rd と val.rd があればそれも利用します。いずれもカレントディレクトリに置いてください。
解析結果は、1ページ分を1行にして画面に表示されます。 左から順に、ページ番号、指定元素の総和、その累計、原子数、 総和の原子当たり平均、累計の原子当たり平均、となります。 画面表示ですので、結果をファイルに保存するには、 適宜リダイレクトしてください。

・rdmov　（最新 Ver.無印＝初期版）
RYUDO の最終構造ファイルである end.rd を読みとり、各原子ごと X,Y,Z 別に計算前後の移動量[m]、および

それらの元素ごとの平均を出力するツールです。基本的に各種表計算ソフトでも容易に可能な処理なので、このツールの必要性は薄いと思いますが、原子数が膨大になる場合には便利かもしれません。
ソースコードをコンパイルする際は、例えば「 gcc rdmov.c -o rdmov 」などとしてください。(math ライブラリは使用していません。)
使い方は、end.rd のあるディレクトリで rdmov を実行するだけです。 結果は画面に表示されますので、ファイルに残したい方は適宜リダイレクトしてください。

(他計算プログラムファイル変換系)

・rdmsp (最新 Ver.1.03a)
TRIBOSIM および MSPORT 用の入力ファイルを RYUDO 用に変換するツールです。
現在は以下の変換が可能です。

file05.dat → def.rd
bondon.dat → bnd.rd

ソースコードをコンパイルする際は、例えば「 gcc rdmsp.c -o rdmsp 」などとしてください。(math ライブラリは使用していません。)
使用方法は、カレントディレクトリに fle05.dat(必要ならば bondon.dat も)をおき、rdmsp を実行するだけです。変換結果は def.rd(および bnd.rd)に保存されます。もし既存の同名ファイルがあった場合は上書きされますのでご注意ください。
bnd.rd は RYUDO 専用の結合情報ファイルです。書式は bondon.dat に比べると非常に簡潔になっており、先頭行に「ryudo-bnd 0.87」と書くほかは、1行1原子として、「原子番号、結合先原子番号1、結合先原子番号2・・・」と記述するだけです。このファイルは、Ver.0.87 以降の RYUDO において、bondon.dat と同等に使用できます。(bonded 行にこのファイル名を指定してください。)
なお、このツールによる file05.dat→def.rd 変換は一部不完全な可能性がありますので、最終的には必ず人の手でチェックするようにしてください。

・rdcar (最新 Ver.1.18)
Accelrys 社製の分子シミュレーションソフトである Cerius2 や MaterialsStudio などから出力される 構造ファイル *.car (および*.mdf) を元に、RYUDO 用の構造ファイル *.rst を作成するツールです。
ソースコードをコンパイルする際は、例えば「 gcc rdcar.c -o rdcar -lm 」などとしてください。(math ライブラリを使用しています。)
使用方法としては、まずカレントディレクトリに *.car ファイルと *.mdf ファイルを置き、rdcar を上記ファイルの * 部分を引数として実行してください。たとえば、abc.car と abc.mdf の場合は、下記のように入力してください。

```
rdcar   abc
```

実行後、自動的に *.rst ファイルが出力されます(上記の例では abc.rst となります)。なお、*.car ファイルは必須ですが、*.mdf ファイルは無くても構いません。*.mdf ファイルが無い場合は、結合情報なしのモデルとして *.rst ファイルを生成します。
car ファイルは、空間群 P1 型(座標展開済み)にのみ対応しています。その他の空間群(座標が折りたたまれている)の car ファイルは変換できませんのでご注意ください。
car ファイルは、単斜晶や三斜晶など、セルの角度が 90 度でないものにも対応しています。この場合、*.rst ファイルも角度はそのままで出力されます(単斜晶や三斜晶の*.rst ファイルになります)。なお、現在の RYUDO 本体は、これら 90 度でないセルに対応していないため、rdset を実行する前に rdrho などを用いて斜方晶(直方体型、全角度が 90 度)セルに変換する必要があります。

carファイルは、周期境界条件が適用されていない(PBC=OFFの)ものにも対応しています。この場合、セルサイズがcarファイル中に記述されていないので、代わりにデフォルトで 100 x 100 x 100 Å が用いられます。また、角度は全て90度になります。

・セルサイズは変更できます。carファイルの第1行（!BIOSYMで始まる行)の中に「CellSize」と追記し、その後ろにx、y、z軸方向別にセルサイズをÅ単位で記述してください。このときCellSizeの前に空白を入れてください。(例:「!BIOSYM」行の最後にスペースを入れてから「CellSize 10.0 10.0 10.0」と追加)

・carファイルの座標をそのままセルに納めた場合、モデルがセルの端に偏ってしまうことがあります。これを修正するため、変換時に座標をずらすことができます(オフセット)。このオフセット機能を利用するには、carファイルの第1行（!BIOSYMで始まる行)の中に「Shift」と追記し、その後ろにx、y、z軸方向別にオフセット量をÅ単位で記述してください。このときShiftの前に空白を入れてください。(例:「!BIOSYM」行の最後にスペースを入れてから「Shift 10.0 -30.0 40.0」と追加)

.rstファイル中の元素名(原子の種類分け)は、通常、.carファイルの各原子名(各原子の座標などを記述している行の第1項)から後ろの数字を取り除いたものになります。もしこれを、FF Type名(同第7項)や、元素名(同第8項)にしたい場合は、*.carファイルの最上行(「!BIOSYM」で始まる行)中に、それぞれ「FFTypeElement」または「StatusElement」と追記してください。また、原子名(第1項)や元素名(第8項)で識別しつつ、FF Type(第7項)で細分化したいときは、「SubFFType」と追記してください(rstファイルにおいて、元素名にFF Type名が追加されるようになります)。

.rstファイル中の分子番号(各原子行の第2項)は通常は全て0になります。.carファイルの分子名(同第5項)＋分子番号(同第6項)に応じて分子番号を変えたい場合は、*.carファイルの最上行(「!BIOSYM」で始まる行)に「NameMolecule」と追記してください。もしくは、各原子記述中に「end」が(1行だけ)出てきたところで分子を区切りたいときは、やはり*.carファイルの最上行に「EndMolecule」と追記してください。なお、これらの分子番号付けは通常0からスタートしますが、もし1からスタートさせたい場合は、*.carファイルの最上行内に「MoleculeFrom1」と追記してください。

*.rstファイルのelem部(各元素種の名称や質量を記述)において、各行末(第4項)にはその元素種に対応するFF Type名が出力されます。ここで、もし1元素に複数のFF Typeがある場合は、カンマ「,」で区切って出力されます。これらの情報をポテンシャルパラメーターの割り当てなどに利用する場合は、できるだけ1つの元素は1つのFF Typeとした方がよいでしょう。このため、たとえばCerius2やMaterialsStudioでモデルを作成する際に原子名をFF Type名以上に細分するか、あるいは本rdcarプログラムでFF Type名を元素の識別に用いる(上述のFFTypeElementを使用するか、SubFFTypeを追加する)とよいでしょう。

結合情報は通常rstファイル内にbond部として出力されます。このとき、各原子記述行の分子内番号(第3項)が原子の通し番号となります。もし結合情報を外部ファイルに出力したい場合は、*.carファイルの最上行内に「BondFileOutput」と記述してください。この場合、*.bndファイルが出力されます。このファイルの書式はbnd.rdと同様です。このファイルをRYUDO計算に使用するにはdef.rdのbonded行でこのファイル名を指定してください。なお、外部ファイルに結合情報を出力する場合は、rstファイル中の各原子の分子内番号は0になります。

注:結合情報をrstファイルのbond部に出力(BondFileOutput非指定)し、かつ、分子名を識別(NameMolecule指定)する際、mdfファイルの結合情報に分子外への参照があると、エラーで停止します。これはrstファイルのbond部に記述できる情報は、同一分子内の結合に限られているためです。

注:RYUDOにおけるset.rdの分子内番号は、現状(Ver.1.21)で65535が上限です。もし原子数が65535個を超える場合は、上記の外部ファイル出力を利用してください。なお、rdcarに限って言えば、65535個を超える場合でもrstファイル内に結合情報を正しく出力できますが、RYUDOはもちろん、その他のプログラム(rdsetやRYUGAなど)でも正常に取り扱えない可能性があります。

rdcarは周期表第3周期までの元素の質量データを内蔵しています。これにより、carファイルの元素名(各原子記述業の第8項)に従って、rstファイルのelem部で自動的に質量が割り当てられます。なお、第4周期以上は質量が0.0になりますので、rdsetにかける前に適宜修正してください。

ソースコード rdcar.c はデフォルトで元素種類数 100、分子数 1000、原子数 10000、元素種類ごとの FF Types 数は 64 まで対応します。もし、それ以上が必要になる場合は、ソースファイル中の定義(#define)を適宜修正してください（元素種は ELEMS、分子数は MOLES、原子数は ATOMS、元素種類ごとの FF Types 数は FFTTS）。

MaterialsStudio の*.car ファイルおよび*.mdf ファイルへの対応については、その書式にまだ分からないことが多く、不完全です。もしお気づきの点がありましたら、具体的な事例をお知らせいただけるとありがたいです。

・rdect　（最新 Ver.無印＝初期版）

一時期の RYUDO（Ver.1.10a～1.12b）が end.rd に出力していた各原子の瞬間被力項（行末3項）を削除するツールです。この時期の RYUDO の end.rd は各原子の記述行が通常よりかなり長くなっており、1行あたり 256 文字の制限があるような既存の解析プログラムでは読み込み時に障害が発生していました。この rdect プログラムを実行すると、カレントディレクトリの end.rd を読み込み、各原子の瞬間被力項を削除して、end1.rd に出力します。副作用として、各原子行の空白文字が最低限になるため、若干見にくくはなりますが、確実に 256 文字未満となるため、既存の各種解析プログラムなどで問題なく読み込めるようになります。

コンパイル方法は下記の通り。特別なライブラリ等は使用していません。

例　gcc rdect.c -o rdect

実行方法は下記の通り。オプション（引数）は必要ありません。

例　./rdect

実行すると、カレントディレクトリの end.rd を読み込み、変換結果を end1.rd というファイル名で出力します。もし既に end1.rd というファイルがカレントディレクトリに存在した場合は、自動的に上書きされますのでご注意ください。

・rd82t87（最新 Ver.1.01（ソース内記述のみ画面出力なし））

def.rd の書式を、Ver.0.82 から Ver.0.87 に変換するツールです。しかし、すでにかなり過去のことなので、現在このツールを使用する機会はほとんど無いと思いますが、一応、情報として残しておきます。

def.rd の Ver.0.82 は RYUDO の Ver.0.82～0.86 で使用されていました（注：RYUDO のバージョンと def.rd のバージョンは基本的に別のものです）。RYUDO は Ver.0.86 から 0.87 にバージョンアップする際、def.rd の結合性ポテンシャル（Ver.0.86 以前では bonded 部）の記述方法を大幅に変更しました。これにより、def.rd のバージョンが 0.82 から 0.87 に変わりました。しかし、結合性ポテンシャル以外はほとんど変更されなかったため、もし結合性ポテンシャルを使用していないならば、def.rd の最上行を「ryudo-def 0.87」に書き換えるだけで、Ver.0.87 相当になります。

ソースコードをコンパイルする際は、例えば「 gcc rd82t87.c -o rd82t87 」などとしてください。（math ライブラリは使用していません。）

使い方としては、変換したい def.rd をカレントディレクトリにおき、rd82t87 を実行するだけです。変換結果は画面に表示されますので、適宜リダイレクトしてファイルに保存してください（例「 rd82t87 > def.new 」）。

このツールは Ver.0.82 より古い（すなわち Ver.0.50 以前の）def.rd には対応していません。ただし Ver.0.50 の def.rd（RYUDOVer.0.50～Ver.0.81 で使用）は、ほとんどの場合「msport」を「bonded」に書き替えるだけで Ver.0.82 相当になります。（実際に rd82t87 で変換する場合、さらに「ryudo-def 0.50」を「ryudo-def 0.82」に書き替えておいてください。なお、def.rd の Ver.0.51～0.81 は存在しません。全て 0.50 と同じです。）

残念ながら、Ver.0.49 以前はさらに異なる書式のため、単純な変換ができません。何とか手作業で変換を行ってください。（もっとも、Ver.0.49 以前の def.rd は内容がかなり少ないため、それほど苦労もなく変換できるとは思います。）

なお、def.rd は人の手でかなり自由に記述できるため、このツールによる自動変換では不十分である可能性もあります。変換結果は最終的に必ず人の手でチェックしてください。

＊ここまでが RYUDO に付属するツール類です。下記は別ソフトとなります。

・RYUGA

各種 MD 計算結果の視覚化と解析を行うための GUI 環境です。RYUDO のバイナリ出力にも対応しています。RYUDO の計算結果の解析には、この RYUGA を推奨します。

RYUGA プログラムの詳細については、RYUGA プログラムの説明書を参照してください。

・RYUKI

各種構造データの加工を行うための GUI 環境です。rst 形式にも対応しています。

詳細は RYUKI プログラムの説明書を参照してください。

RYUDOの権利関係

著作権法を始めとする関係法令に照らし合わせて、当方では下記のように判断します。

・RYUDOについて

　RYUDOは、三浦隆治が宮本研究室の学生時代に研究目的で開発したものです。学生ですので、法人としての東北大学（宮本研究室）の業務に従事する者ではありません。よって、RYUDOの著作者は個人としての三浦隆治となります（著作権法第十五条）。また資料や資材などの提供をしただけでは著作者になることができませんので（著作権法第二条一項二号および判例等による）、他のいずれの個人、団体ともRYUDOの著作者には含まれていません。よって、RYUDOの著作者人格権は三浦隆治が単独で有しています。この著作者人格権は、いかなる場合も譲渡されません（著作権法第五十九条）。

　著作者人格権（著作権法第二章第二款）には、公表権、氏名表示権、同一性保持権が含まれています。三浦隆治は公表権（著作権法第十八条）に基づき、RYUDOをフリーソフトウェアとして公開するよう規定します。また氏名表示権（著作権法第十九条）に基づいて三浦隆治の著作であることを明記するよう規定し、同一性保持権（著作権法第二十条）に基づいて無許可の改変を禁じています。

　著作権（財産権）（著作権法第二章第三款）については、現在も全て三浦隆治が有しています。ただし、法人としての東北大学（宮本研究室）には、全ての権利を無条件で許諾しています。同様に、一般のRYUDO利用者に対しても先述の通り、一部の権利を条件付きで許諾しています。（著作権法第六十三条）

・NEW-RYUDOについて

　NEW-RYUDOは、法人としての東北大学（宮本研究室）が、RYUDOをベースに開発したプログラムで、個人としての三浦隆治との共同著作物になります。その著作者人格権および著作権（財産権）の行使については、三浦隆治は全て宮本研究室に一任しています（著作権法第六十四条および第六十五条）。

・NEW-RYUDO発表後の、RYUDOおよびNEW-RYUDOの改良、機能追加について

　RYUDOの著作者である三浦隆治は、翻訳・翻案権（著作権法第二十七条）に基づき、現在も個人の趣味としてRYUDOの改良と機能追加を行っています。これらの開発はNEW-RYUDOとは独立しているため、その著作権は全て個人としての三浦隆治が有しています。そしてそれらの新機能は、その著作者（個人としての三浦隆治）の許諾を得て、法人としての東北大学（宮本研究室）の職員（三浦隆治）が、業務としてNEW-RYUDOに移植していますので、NEW-RYUDOでも同様の新機能を利用することができます。

　なお、NEW-RYUDOにおいては、RYUDOとは独立して改良と機能追加を行っている部分があります（プロトン伝導計算機能など）。これは東北大学（宮本研究室）の職員（三浦隆治、他）が業務として開発したものですので、RYUDOにはありません（本マニュアルにも含まれていません）。